Collaborative Scheduling and Routing of Home Healthcare Service across Multiple Communities

Yang Wang[1], Wenjie Hu[1], Jin-Kao Hao[2], Jianguang Feng[1*]

[1*]School of Management, Northwestern Polytechnical University, Xi'an, 71072, China.
[2]Department of Computer Science, Université d'Angers, Angers, 49045, France.

*Corresponding author(s). E-mail(s): feng@nwpu.edu.cn;
Contributing authors: yangw@nwpu.edu.cn; huwenjie@mail.nwpu.edu.cn; jin-kao.hao@univ-angers.fr;

Abstract

As an important home health care mode in China, the home-community service mode focuses on the impact of healthcare centers in each community and coordinates the medical resources between communities. In this paper, we propose a new collaborative home health care routing and scheduling problem in multiple service centers (C-HHCRSP). In addition to the traditional routing and scheduling decisions, C-HHCRSP also assigns a working center for each highly qualified caregiver who can provide health services across multiple communities. This problem is challenging due to the presence of complex, realistic constraints such as time windows, mandatory lunch breaks, synchronized visits, and downgrading of services. We first formulate the problem as a mixed integer programming model, which is solved by CPLEX. Then, we propose an adaptive large neighborhood search (ALNS) algorithm that integrates new problem-specific destroy and repair operators. To further improve the performance of ALNS, we propose two post-optimization techniques, which are based on heuristic strategies and a set partitioning model, respectively, resulting in the enhanced algorithms ALNS-HS and ALNS-SP. Tested on 104 benchmark instances, ALNS-HS shows a competitive performance with CPLEX on small instances. For large instances, CPLEX fails to find feasible solutions, and we compare our heuristic algorithms with a classic large neighborhood search algorithm to demonstrate their superiority. Additional analyses are performed to verify the roles of the components of the proposed algorithms.

Keywords: Home health care, Multiple healthcare centers, Cooperative scheduling, Large neighborhood search, Post-optimization techniques

1 Introduction

In recent years, the aging of the population has become a global issue, with China experiencing a continuous acceleration of this phenomenon. In 2018, the population aged 60 and over reached 249 million, representing 17.9% of the total population. By 2021, this number will exceed 267 million, accounting for 18.9%. At this rate, China will even become a super-elderly country by 2025. The continuous growth of the elderly population has put unprecedented pressure on China's healthcare system, leading to significant strain on limited medical resources. To address this challenge, Home Health Care (HHC) has gained increasing attention and popularity. Unlike traditional medical models, HHC involves healthcare centers that dispatch qualified caregivers to provide medical treatment or nursing services in patients' homes at scheduled times. This approach allows patients to receive medical care in a familiar environment, eliminating the need to travel for medical services. Caregivers are a crucial resource in HHC, and the efficiency of their dispatching has a direct impact on the quality of home healthcare services.

In general, caregivers depart from a health center to provide services to patients distributed in different areas for injections, medication, etc., and return to the center upon completion of all services. This process necessitates the consideration of several factors, including aligning the patient's needs with the caregiver's qualifications, respecting the patient's preferences and balancing the caregiver's workload. To optimize the allocation of medical resources, the Home Health Care Routing and Scheduling Problem (HHCRSP) is an important problem that has been widely studied in the literature. HHCRSP is mainly regarded as an extension of the vehicle routing problem (VRP) with time windows and multiple depots [18]. Additional features related to home health care are considered in HHCRSP as well. Due to the complexity introduced by VRP and these additional features, HHCRSP is an NP-hard problem [40]. Thus, finding an optimal solution for HHCRSP is computationally challenging and heuristic-based methods are commonly used to solve such problems [2, 52].

However, there are many densely populated communities in China, and these communities are required to assume a critical role in HHC by the government. In general, each community is equipped with one healthcare center and various medical resources. Considering the shortage of certain medical resources, especially the key resources (e.g., highly qualified caregivers, technical devices), the decision-maker needs to coordinate these resources across communities to meet the needs of patients. Therefore, the home-community service mode has become a crucial home health care mode to improve the responsiveness of healthcare centers to patients' requirements and to enhance the ability to integrate medical resources between communities. To properly address the realistic requirement, we introduce a new collaborative home health care routing and scheduling problem in multiple service centers (C-HHCRSP). In addition to determining the set of patients each caregiver serves and building their visiting routes, C-HHCRSP needs to coordinate the highly qualified caregivers among communities and assign them to appropriate healthcare centers. Furthermore, C-HHCRSP also considers other complex, realistic constraints, including time windows, mandatory lunch breaks, synchronized visits, downgrading of services, etc.

To the best of our knowledge, this is the first HHCRSP study considering these additional features. The main contributions of this paper are summarized as follows.

- Multiple healthcare centers and more comprehensive realistic constraints are considered in C-HHCRSP. A mixed integer programming model is developed to formulate the problem, which can be solved by the CPLEX solver to obtain satisfying results for small instances.
- An adaptive large neighborhood search (ALNS) algorithm is proposed to find effective solutions for large problem instances. Seven new operators are developed in ALNS, including the destroy operators specially designed to assign the caregiver's working center and the repair operators for the synchronized visits and lunch break constraint. They could help the algorithm explore a larger solution space and find better solutions of high quality.
- Two effective post-optimization techniques, based on the heuristic strategy and the set partitioning, are developed to enhance the algorithm's performance within a short computing time. The heuristic strategy includes two heuristic procedures, which are performed sequentially. In the set partitioning, a problem-specific set partitioning model is formulated, aiming to create a high-quality solution based on the routes found by ALNS. Combining these two post-optimization techniques with ALNS, we can obtain ALNS-HS and ALNS-SP, respectively.
- We present 9 sets of 104 benchmark instances with different characteristics and use them to evaluate the proposed model and algorithm. We compare ALNS-HS with the CPLEX solver and conclude that ALNS-HS performs as well as the CPLEX solver on small instances. We then compare our ALNS-HS and ALNS-SP algorithms with a classical large neighborhood search (LNS) on large instances. We also perform additional analyses to evaluate the impact of post-optimization techniques, the scheduling of lunch breaks, and all the proposed destroy/repair operators.

The remainder of the paper is organized as follows. A review of the literature is given in Section 2. A detailed description of the problem and its formulation is provided in Section 3. Section 4 presents our solution method, and Section 5 shows the generation of instances and the computational results. Finally, Section 6 contains the conclusions.

2 Literature review

To our knowledge, HHCRSP was first proposed by Fernandez et al [16]. We will review the literature regarding the objectives and constraints introduced in HHCRSP and the applied solution methods. Some representative research works are summarized in Table 1.

For HHCRSP, routing cost, preferences of patients and caregivers, workload balance, etc., are popular optimization objectives in the literature. Most studies model HHCRSP as an extension of the vehicle routing problem and use the routing cost as the objective. Holm and Angelsen [28] showed that the driving time spent by caregivers accounted for 18%–26% of their entire working time. This means there is great potential to optimize the routing of caregivers to improve operational costs and reduce

Table 1 Related work on HHCRSP

Paper	Objective(s)				Constraints					Solution approach(es)
	RC	PF	VC	WB	TW	SK	WT	LB	SS	
Exact methods										
Rasmussen et al [42]	✓	✓	✓		✓	✓	✓		✓	Branch-and-price
Trautsamwieser and Hirsch [51]				✓	✓	✓	✓	✓		Branch-price-and-cut
Cappanera and Scutellà [7]				✓	✓	✓	✓			Integer linear programming
Yuan et al [54]	✓		✓		✓	✓				Branch-and-price
Liu et al [30]	✓		✓		✓		✓	✓		Branch-and-price
Heching et al [24]	✓				✓	✓	✓			Logic-based Benders decomposition
Fathollahi-Fard et al [15]	✓				✓					Lagrangian relaxation
Hashemi Doulabi et al [23]	✓		✓	✓	✓		✓		✓	Branch-and-cut
Yin et al [53]	✓	✓			✓	✓	✓			Branch-and-price-and-cut
Heuristic methods										
Milburn and Spicer [36]	✓			✓			✓			Tabu search
Bard et al [4]	✓				✓	✓	✓	✓		Greedy randomized adaptive search
Mankowska et al [32]	✓		✓		✓	✓			✓	Adaptive variable neighborhood search
Mısır et al [37]	✓		✓	✓	✓	✓	✓			Hyper-heuristics
Redjem and Marcon [43]	✓				✓	✓	✓		✓	Caregivers Routing Heuristic
Rest and Hirsch [44]	✓		✓		✓	✓	✓	✓		Tabu search
Decerle et al [12]	✓		✓		✓	✓	✓		✓	Memetic algorithm
Ait Haddadene et al [1]	✓	✓			✓	✓			✓	Non-Dominated Sorting Genetic Algorithm
Nikzad et al [41]	✓		✓		✓	✓	✓			Metaheuristics
Liu et al [31]	✓				✓	✓	✓	✓	✓	Metaheuristics
Masmoudi et al [33]	✓	✓		✓	✓		✓		✓	Metaheuristics
Fu et al [19]	✓		✓		✓	✓				Artificial bee colony algorithm
Combination of heuristic and exact methods										
Barrera et al [5]	✓			✓			✓			Mathematical approach/route packing
Hiermann et al [27]			✓		✓	✓	✓			Constraint programming/Metaheuristics
Riazi et al [45]	✓				✓	✓	✓			Gossip algorithm/column generation
Grenouilleau et al [21]	✓		✓		✓	✓	✓			LNS/set partitioning
Chen et al [9]	✓		✓		✓	✓	✓			LNS/set partitioning
This paper	✓		✓		✓	✓	✓	✓	✓	ALNS/post-optimization

RC: Routing cost; PF: Preference; VC: Violation of constraints; WB: Workload balance; TW: Time windows; SK: Skill; WT: Worktime; LB: Lunch break
SS: Synchronized services

expenses. Mendoza-Alonzo et al [35] also showed that as the size of the optimization problem increased, the caregiver's route cost became closer to the total cost. The preference of patients and caregivers affects the service quality. Rasmussen et al [42] set a preference parameter that gave a negative cost if caregivers wanted to serve the patients or a positive cost if they did not. Workload balance is often used as a main measure of the caregiver's satisfaction. For example, the objective of minimizing the maximum working hours of the caregiver was adopted in [36, 40, 48]. Other measures of workload balance include minimizing the difference between maximum and minimum working hours of all caregivers used in [5] and an integrated evaluation of travel load, case load, and visit load used in [26].

Time windows, skill requirements, and working time regulations are common constraints considered in most HHCRSP models. Given that many services in home health care are very sensitive to time, most studies took the time window of the visit required by patients as a hard constraint [4, 43]. For example, insulin injection or drug supply must be completed within a specific time window. Some studies also set the time window of the visit as a soft constraint [32, 54], which can be violated but induces a penalty cost. Regarding skill requirements, Hiermann et al [27] divided caregivers into five types, and each type of caregiver can only perform tasks with weaker qualifications. Fikar and Hirsch [17] limited the maximum number of downgrades to reduce the overqualified service and ensure more efficient use of resources. Working time regulations are usually implemented by setting a working time window or a limit of working days or duration for each caregiver [13, 38, 39]. The lunch break of caregivers has a great impact on their satisfaction, but few studies have addressed this feature. Bachouch et al [3] allowed caregivers to have one hour per day for lunch and assigned a break node as a fictitious care to each caregiver's route. On this basis, Rest and Hirsch [44] allowed the caregiver's break time to be split and allocated within the working day. In addition, some papers focused on considering synchronization constraints. According to Soares et al [49], synchronization aspects can be categorized into two categories: operation synchronization and movement synchronization. Operation synchronization involves requirements on the execution time of interdependent tasks on different routes, whereas movement synchronization pertains to requirements on the execution sequence of interdependent tasks on different routes. In the HHCRSP literature, the majority of studies focus on operation synchronization, which mandates that the start service time of caregivers on synchronized visits should be simultaneous [6, 23, 42]. To the best of our knowledge, few studies have concurrently addressed the aforementioned constraints, with Liu et al [31] being a notable exception. They introduced a novel formulation of the HHCRSP, denoted as HHCRSPsynLB, which comprehensively integrates considerations such as time windows, synchronized visits, and lunch breaks.

From a methodological point of view, HHCRSP is mainly regarded as an extension of VRP. Exact algorithms have been widely designed to obtain optimal solutions for small problem instances. For example, Yuan et al [54] designed a branch and pricing algorithm for HHCRSP with stochastic service times and skill requirements, where a column generation algorithm was designed to solve the master problem, and a labeling algorithm with some acceleration techniques was used to solve the pricing subproblem.

In Rasmussen et al [42], the Home Care Crew Scheduling Problem was considered a set partitioning problem with side constraints and a branch pricing method was proposed. Temporal dependencies were considered as a generalized precedence constraint and enforced by branching. Cappanera and Scutellà [7] applied integer linear programming techniques to formulate a weekly planning problem where the concept of the pattern was introduced as a key tool to address assignment, scheduling and routing decisions jointly. Valid inequalities and cuts were added to enhance the models. Rodriguez et al [46] developed a two-stage stochastic programming approach to determine the dimension of caregivers in homecare services, taking into account the uncertainty of historical demand, the territory model and the amount of time for treatment. Carello and Lanzarone [8] designed a cardinality-constrained robust model for the nurse-to-patient assignment problem in HHC services. This model was easily applied to account for the uncertain patient demands without assuming probability distributions or deriving scenarios.

To handle large problem instances, heuristic algorithms have proved to be more effective. Shao et al [48] proposed a parallel greedy randomized adaptive search algorithm to solve the therapist routing and scheduling problem, where the first phase constructed the daily routes for the therapists to form weekly schedules, and the second phase employed neighborhood search to improve the solution quality further. Mankowska et al [32] proposed an adaptive variable neighborhood search (AVNS) algorithm for the daily planning of HHC that considered patient's requirements, caregiver's qualifications and interdependencies of services. AVNS started from an initial solution found by a greedy heuristic and jointly used eight different neighborhood structures for solution improvement. Grenouilleau et al [21] proposed a set partition heuristic method considering realistic constraints, which combined the set partition formulation with the large neighborhood search framework. A column generated by the large neighborhood search algorithm was first used to solve the linear relaxation problem of the set partitioning model, which was followed by a constructive heuristic to generate integer solutions. Chen et al [9] built a bi-level programming model for HHCRSP considering the travel time uncertainty and designed a three-stage hybrid algorithm to solve the model. The first stage was based on the iterated local search framework to generate high-quality routes, while the second stage iteratively solved a set partitioning model, and the third stage used a post-optimization method to improve the solution further. Hiermann et al [27] designed a two-stage approach for a real-world multimodal home-healthcare scheduling problem. In the first stage, constraint programming techniques and a random construction procedure were used to generate an initial solution. In the second stage, one of the four metaheuristics, including variable neighborhood search, a memetic algorithm, scatter search and simulated annealing hyper-heuristics, was selected respectively to improve the initial solution. Fu et al [19] introduced a multi-objective artificial bee colony algorithm for the HHCRSP with a sharing strategy among multiple HHC centers. Their approach integrated problem-specific knowledge-based techniques for population initialization and solution refinement, significantly augmenting the algorithm's efficacy.

Among the various heuristic algorithms, the ALNS algorithm has been successfully applied to solve a wide range of routing and scheduling problems. Its capacity to improve solutions through iterative destroy and repair operations aligns well with the nature of such problem. Yazır et al [52] developed an ALNS metaheuristic for the multi-period HHCRSP that considered homogeneous electric vehicles and time windows. The ALNS incorporated a specially designed 10-step sequential neighborhood change procedure, along with five destroy and four repair operators, rendering the algorithm robust and effective. Erdem and Koç [14] proposed a hybrid ALNS algorithm to address the multi-depot HHCRSP considering electric vehicles and private and public charging stations. Construction heuristic and variable neighborhood descent-based local search procedures were integrated in the hybrid ALNS algorithm. By combining existing heuristic mechanisms with new problem-specific procedures, this algorithm was capable of achieving high-quality solutions.

3 Problem description

3.1 Problem definition

Given a set of patients distributed in different communities and a set of caregivers, C-HHCRSP aims to assign highly qualified caregivers to healthcare centers, determine the patients each caregiver serves, and build the visit route of each caregiver over a one-day planning horizon. Caregivers can be divided into general and professional ones according to their skills. Each general caregiver lives in the community with a fixed daily working healthcare center and provides services to patients within his/her community. In contrast, professional caregivers are qualified to provide high-level care services and often serve as a key and scarce medical resource. Their working healthcare centers are not fixed, requiring appropriate assignment. To guarantee the full utilization of these essential resources and prevent over-concentration in certain areas and shortages in others, it is vital to allow these professional caregivers to work in different communities. This also ensures that healthcare services are delivered efficiently and effectively to the patients in need. The two types of caregivers may adopt different means of transport (e.g., car, electric bicycle). Therefore, the route costs of the two transport modes per unit travel distance are different. Caregivers should work within their time window and cannot start work in advance. Due to the direct impact of overtime on caregiver satisfaction, additional costs are incurred in the objective function when overtime occurs. Moreover, caregivers must take a lunch break at their working healthcare center during the lunchtime window.

Each patient needs to be served exactly once a day, and these visits are divided into three types. Type-I visits are general visits that do not require a high level of skill and can only be performed by general caregivers with qualified skills. A certain degree of downgrading is allowed when determining which patients should be served by general caregivers, meaning that overqualified caregivers will perform some general visits. Type-II visits require a high level of skill and can only be performed by professional caregivers. Type-III visits are synchronized visits, which need a professional caregiver and a general caregiver to serve the patients simultaneously. Each service has a prescribed time window indicating the preferred service time. Considering that timely healthcare service is an important factor in patient satisfaction, the objective function will integrate penalty costs for service delays.

Figure 1 illustrates an example of C-HHCRSP. Squares 1–4 represent community healthcare centers, the blue circles 1–18 are type-I visits, the yellow circles 19–21 are type-II visits, and the green circles 22–24 are type-III visits. The routes traversed by general and professional caregivers are represented by black and red arrows, respectively. The dotted box represents the service scope of communities. Take the routes traversed by caregivers A and D as an example. Caregiver A departs from Community 1, performs visits 19, 22, and 20 in turn and returns to the community. Caregiver D departs from Community 1, performs visits 4, 22, and 5 in turn and returns to the community. Visit 22 is a synchronized visit that needs caregivers A and D to serve the patient simultaneously.

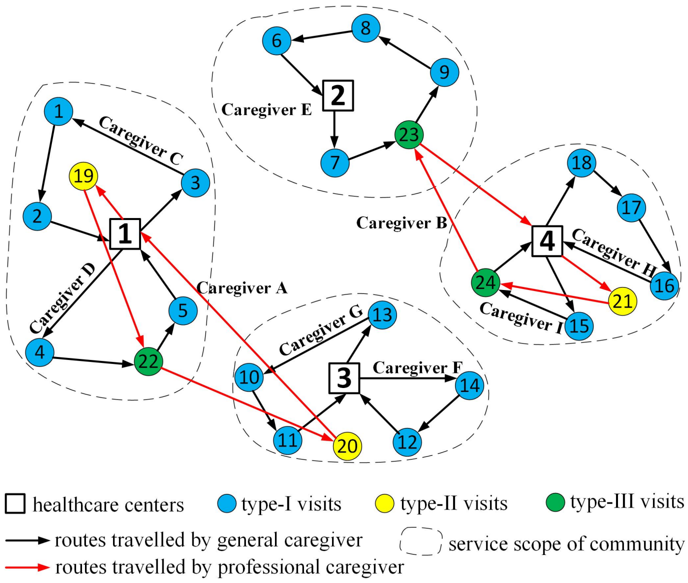

Fig. 1 An example of the C-HHCRSP

3.2 Problem formulation

We model C-HHCRSP on a digraph $G = (V, A)$. $V = V_D \cup V_C$ is the set of vertices, where $V_D = \{1, \ldots, |V_D|\}$ denotes the set of community healthcare centers and $V_C = \{|V_D| + 1, \ldots, |V_D| + |V_C|\}$ denotes the set of patients. $A = \{(i, j) \mid i, j \in V_D \cup V_C, (i, j) \notin V_D \times V_D\}$ is the set of arcs connecting community healthcare centers and patients. Let d_{ij} denote the travel distance of edge $(i, j) \in A$, and $d_{ij} = d_{ji}$. We consider a set of patients P, where each patient $i \in P$ requires a daily visit. Each visit is characterized by a soft time window $[SDL_i, SDU_i]$, a service duration D_i and a required skill level q_i^R. QN is the required skill level threshold, where type-I visits are those with $q_i^R \leq QN$ and type-II visits are those with $q_i^R > QN$. Let K be the set of caregivers needed to complete the services. For each caregiver $k \in K$, a skill level q_k is specified, in addition to a working time window $[WL_k, WU_k]$, a lunch break time window $[LL_k, LU_k]$ and a lunch break duration t_k^{Lun}. The sets of professional and general caregivers are denoted by K_1 and K_2, respectively. Each caregiver departs from his/her working healthcare center and returns to it after performing all visits. The objective is to determine the set of patients each caregiver serves and the corresponding visiting route. For professional caregivers, we should assign them to appropriate healthcare centers as well. A mixed integer programming (MIP) model is formulated, and the notation used in the model is summarized in Table 2.

Table 2 Notation

Sets and parameters	
V	Set of vertices, $V = V_D \cup V_C$. V_D The set of community healthcare centers, where each healthcare center h is represented by h^f and h^s, h^f denotes the origin and h^s denotes the destination; V_C The patient set, $V_C = \cup_{h\in V_D} V_C^h$
V_C^h	The patient set within the service scope of community healthcare center h
P^{syn}	The set of synchronized visits, a patient who requires synchronized visits can be seen as two identical patients $i, j \in V_C$, there is $(i,j) \in P^{syn}$
K	The set of caregivers, $K = K_1 \cup K_2$, K_1 The set of professional caregivers, K_2 The set of general caregivers, $K_2 = \cup_{h\in V_D} K_2^h$
K_2^h	The set of general caregivers working in the community h
t_{ij}	The travel time between vertex i and j, where $i, j \in V$
d_{ij}	The travel distance between vertex i and j, where $i, j \in V$
ξ_1, ξ_2	The unit travel cost coefficient of professional and general caregivers
D_i	The service duration of visit required by patient $i \in V_C$
q_i^R	The skill level of visit required by patient $i \in V_C$
q_k	The skill level of caregiver $k \in K$
QN	The skill level threshold to distinguish type-I and type-II visits
$[SDL_i, SDU_i]$	The soft time window of visit required by patient $i \in V_C$
$[WL_k, WU_k]$	The working time window of caregiver $k \in K$
$[LL_k, LU_k]$	The lunchtime window of caregiver $k \in K$
t_k^{Lun}	The lunch break duration of caregiver $k \in K$
C^p	The unit penalty cost of delays for visits
C^g	The unit penalty cost of overtime for caregivers
Decision variables	
x_{ij}^k	Binary variable that is 1 if caregiver k travels from i to j, 0 otherwise
z_{ij}^k	Binary variable that is 1 if caregiver k takes the lunch break after serving patient i and before traveling to patent j, 0 otherwise
y_h^k	Binary variable that is 1 if professional caregiver k is assigned to community h, 0 otherwise
ts_i^k	Starting time when caregiver k works at vertex i
u_i	Delay of serving patient i
o_k	Overtime performed by caregiver k

$$\max\ f = \alpha \sum_{i\in V}\sum_{j\in V} d_{ij}\left(\sum_{k\in K_1} x_{ij}^k \xi_1 + \sum_{k\in K_2} x_{ij}^k \xi_2\right) + \beta C^p \sum_{i\in V_C} u_i + \gamma C^g \sum_{k\in K} o_k \tag{1}$$

$$\text{s.t.} \sum_{j\in V_C^h} x_{h^f j}^k = \sum_{j\in V_C^h} x_{jh^s}^k = 1 \quad \forall h \in V_D, k \in K_2^h \tag{2}$$

$$\sum_{h\in V_D}(y^k_{h^f}+y^k_{h^s})=2 \quad \forall k\in K_1 \tag{3}$$

$$y^k_{h^f}=y^k_{h^s} \quad \forall h\in V_D, k\in K_1 \tag{4}$$

$$\sum_{j\in V_C}x^k_{h^fj}=\sum_{j\in V_C}x^k_{jh^s}=y^k_{h^f} \quad \forall h\in V_D, k\in K_1 \tag{5}$$

$$\sum_{j\in V}x^k_{ij}=\sum_{j\in V}x^k_{ji} \quad \forall i\in V_C, k\in K \tag{6}$$

$$\sum_{j\in V^h_C\cup h}\sum_{k\in K^h_2}x^k_{ij}=1 \quad \forall h\in V_D, i\in V^h_C|q^R_i\le QN \tag{7}$$

$$\sum_{j\in V}\sum_{k\in K_1}x^k_{ij}=1 \quad \forall i\in V_C|q^R_i>QN \tag{8}$$

$$SDL_i\sum_{j\in V}x^k_{ij}\le ts^k_i\le SDU_i+u_i+(1-\sum_{j\in V}x^k_{ij})M \quad \forall i\in V_C, k\in K \tag{9}$$

$$ts^k_i+D_i+t_{ij}+(x^k_{ij}-1)M\le ts^k_j \quad \forall i,j\in V, k\in K \tag{10}$$

$$q^R_i\le\sum_{j\in V}\sum_{k\in K_2}x^k_{ij}q_k \quad \forall i\in V_C|q^R_i\le QN \tag{11}$$

$$1+q^R_i\ge\sum_{j\in V}\sum_{k\in K_2}x^k_{ij}q_k \quad \forall i\in V_C|q^R_i\le QN \tag{12}$$

$$ts^k_i\ge WL_k\sum_{j\in V}x^k_{ij} \quad \forall i\in V_C, k\in K \tag{13}$$

$$ts^k_i+D_i\le WU_k+o_k+(1-\sum_{j\in V}x^k_{ij})M \quad \forall i\in V_C, k\in K \tag{14}$$

$$WL_k\le ts^k_{h^f}\le ts^k_{h^s}\le WU_k+o_K \quad \forall h\in V_D, k\in K^h_2 \tag{15}$$

$$(y^k_{h^f})WL_k\le ts^k_{h^f}\le ts^k_{h^s}\le(y^k_{h^f})WU_k+o_K \quad \forall h\in V_D, k\in K_1 \tag{16}$$

$$\sum_{i\in V^h_C\cup h}\sum_{j\in V^h_C\cup h}z^k_{ij}=1 \quad \forall h\in V_D, k\in K^h_2 \tag{17}$$

$$\sum_{i\in V}\sum_{j\in V}z^k_{ij}=1 \quad \forall k\in K_1 \tag{18}$$

$$z^k_{ij}\le x^k_{ij} \quad \forall k\in K, i,j\in V \tag{19}$$

$$ts^k_j\ge ts^k_i+D_i+t_{ih^s}+t^{Lun}_k+t_{h^fj}+M(z^k_{ij}-1) \quad \forall h\in V_D, k\in K^h_2, i,j\in V^h_C\cup h \tag{20}$$

$$ts^k_j\ge ts^k_i+D_i+t_{ih^s}+t^{Lun}_k+t_{h^fj}+M(z^k_{ij}+y^k_{h^f}-2) \quad \forall h\in V_D, k\in K_1, i,j\in V_C\cup h \tag{21}$$

$$ts^k_i+D_i+t_{ih^s}\ge LL_k+M(\sum_{j\in V^h_C\cup h}z^k_{ij}-1) \quad \forall h\in V_D, k\in K^h_2, i\in V^h_C\cup h \tag{22}$$

$$ts_i^k + D_i + t_{ih^s} \leq LU_k - t_k^{Lun} + M(1 - \sum_{j \in V_C^h \cup h} z_{ij}^k) \quad \forall h \in V_D, k \in K_2^h, i \in V_C^h \cup h \tag{23}$$

$$ts_i^k + D_i + t_{ih^s} \geq LL_k + M(y_{h^f}^k + \sum_{j \in V_C \cup h} z_{ij}^k - 2) \quad \forall h \in V_D, k \in K_1, i \in V_C \cup h \tag{24}$$

$$ts_i^k + D_i + t_{ih^s} \leq LU_k - t_k^{Lun} + M(2 - y_{h^f}^k - \sum_{j \in V_C \cup h} z_{ij}^k) \quad \forall h \in V_D, k \in K_1, i \in V_C \cup h \tag{25}$$

$$\sum_{k \in K} ts_i^k = \sum_{k \in K} ts_j^k \quad \forall (i,j) \in P^{syn} \tag{26}$$

$$\sum_{m \in V} \sum_{k \in K_1} x_{im}^k = \sum_{n \in V} \sum_{l \in K_2} x_{jn}^l = 1 \quad \forall (i,j) \in P^{syn} \tag{27}$$

$$y_h^k \in \{0,1\} \quad \forall h \in V_D, k \in K_1 \tag{28}$$

$$x_{ij}^k \in \{0,1\} \quad \forall i,j \in V, k \in K \tag{29}$$

$$ts_i^k \geq 0 \quad \forall i \in V, k \in K \tag{30}$$

$$u_i \geq 0 \quad \forall i \in V_C \tag{31}$$

$$o_k \geq 0 \quad \forall k \in K \tag{32}$$

The objective function (1) is to minimize the total weighted cost, including travel costs, penalty costs for deviations from preferred visit times and overtime costs. Constraint (2) ensures that general caregivers depart from and return to their working healthcare center after visiting all the assigned patients. Constraints (3) and (4) require that each professional caregiver is assigned to one healthcare center per day. Constraint (5) ensures that professional caregivers depart from and return to their working healthcare center after visiting all the assigned patients. Constraints (6) state that the flow conservation balance, meaning that a caregiver should leave a patient after the service. Constraints (7) ensure that general caregivers can perform general visits within the same community. Constraints (8) ensure that each type-II visit should be served by a professional caregiver once. Constraints (9) indicate the upper soft time window violations of each visit. Constraints (10) ensure that the travel time of caregivers between two consecutive visits must be respected. Constraints (11) and (12) indicate that general visits can only be performed by general caregivers with qualified skill levels, where degraded service with a maximum deviation of 1 is allowed. Constraints (13)–(16) indicate that each caregiver should work within the given time window, and overtime is possible. Constraints (17)–(19) enforce that each caregiver needs to have a lunch break. Constraints (20) and (21) ensure that caregivers have enough time to take a lunch break at healthcare center after finishing visit i, and then continue to perform visit j. Constraints (22) and (24) indicate that each caregiver's lunch break must start within the prescribed lunchtime window. Constraints (23) and (25) ensure that caregivers must finish their lunch break strictly during the lunchtime window. Constraints (26) and (27) ensure that the synchronized visits must be served by a general caregiver and a professional caregiver simultaneously. Constraints (28)–(32) define the domains of the decision variables.

Figure 2 illustrates a C-HHCRSP with 2 healthcare centers, 3 caregivers and 12 patients. The three tables present the input data for the instance, with symbols corresponding to the definitions provided in Table 2. For example, the caregiver with ID 3 is a professional caregiver with an unassigned working center, skill level of 5, working time window of [0, 1200], lunchtime window of [400, 600], and lunch break duration of 30. Patient with ID 5 belongs to healthcare center 1, requiring a skill level of 5, indicating synchronized visits. The service time window is [510.30, 552.25], with a duration of 63.24. The optimal solution determines the working center for each professional caregiver, the set of patients served by each caregiver, and their corresponding visiting route. The optimal objective function value is $f = 328.28$. Circles 1–12 represent the visits required by patients, with the numbers in square brackets above the circle representing the start service time for each visit. Squares 1–2 represent community healthcare centers. Since visit 5 is a synchronized visit, both the professional caregiver and general caregiver 1 should serve it simultaneously.

Instance

The sets and parameters related to caregivers

ID	Working center	q_k	$[WL_k, WU_k]$	$[LL_k, LU_k]$	t_k^{Lun}
1	1	2	[0, 1200]	[400, 600]	30
2	2	2	[0, 1200]	[400, 600]	30
3	unassigned	5	[0, 1200]	[400, 600]	30

The sets and parameters related to healthcare centers

Healthcare center	X-coord	Y-coord
1	-4.09	-6.64
2	3.42	9.22

The sets and parameters related to patients

ID	X-coord	Y-coord	Center	q_i^R	$[SDL_i, SDU_i]$	D_i
1	1.81	-0.36	1	1	[867.99, 911.89]	46.89
2	-8.70	-9.48	1	4	[426.01, 493.51]	60.55
3	-4.55	-6.19	1	1	[748.58, 806.52]	64.97
4	-6.16	-5.67	1	2	[1161.45, 1214.02]	63.30
5	0.23	-12.50	1	5	[510.30, 552.25]	63.24
6	-7.18	-5.64	1	2	[849.93, 901.80]	56.85
7	4.79	8.86	2	1	[642.94, 702.44]	57.41
8	3.98	10.20	2	5	[461.27, 514.98]	58.82
9	3.21	9.91	2	2	[322.19, 370.04]	53.35
10	0.65	11.68	2	2	[261.28, 319.68]	59.10
11	6.44	8.53	2	1	[1022.98, 1084.28]	64.53
12	1.44	6.14	2	1	[174.21, 219.62]	48.11

Optimal Solution: $f = 328.28$

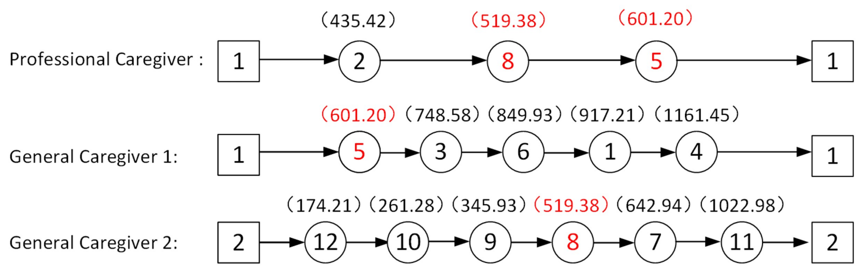

Fig. 2 An example of the C-HHCRSP with parameters and optimal solution

To the best of our knowledge, few studies have considered such a complex problem. While both our work and the HHCRSPsynLB proposed by Liu et al [31] address routing and scheduling challenges in the home health care domain, several significant distinctions exist in the specific constraints considered. First, in HHCRSPsynLB, only one healthcare center is accounted for, allowing caregivers to return to the same center after serving patients. In contrast, C-HHCRSP considers multiple healthcare centers and categorizes caregivers into general and professional classes. Professional caregivers are mandated to depart from and return to specific centers, while general caregivers serve within their designated communities. Second, in HHCRSPsynLB, caregivers take lunch breaks at patient locations. Conversely, C-HHCRSP mandates lunch breaks at working healthcare centers. Third, while both models incorporate synchronized visits, C-HHCRSP specifies that such visits must involve a professional and a qualified general caregiver simultaneously. Finally, C-HHCRSP introduces a maximum deviation constraint to mitigate overqualified service, promoting resource efficiency. Time windows in C-HHCRSP are treated as soft constraints, allowing service delay with penalty costs, whereas HHCRSPsynLB treats them as hard constraints.

4 Solution method

In this section, we first present a two-stage greedy insertion algorithm to obtain an initial solution based on the characteristics of the problem. Then, we develop an ALNS algorithm that includes well-designed destroy and repair operators to obtain local optimal solutions. Two post-optimization techniques are proposed to improve the solution obtained by ALNS. By combining the ALNS algorithm with the post-optimization techniques, we obtain the algorithms ALNS-HS and ALNS-SP.

4.1 Main scheme

ALNS is a powerful meta-heuristic that improves solutions by iteratively performing destroy and repair operations and has been used to effectively solve various traffic and scheduling problems [22, 29, 34]. The ALNS algorithm can achieve high-quality and feasible solutions within a reasonable computational time by incorporating local search operators. Additionally, employing post-optimization techniques that maintain the majority of the solution's structure while making minor adjustments can effectively enhance the algorithm's performance. Thus, we integrate the ALNS algorithm with post-optimization techniques for effectively solving C-HHCRSP.

Algorithm 1 presents the proposed algorithms. First, a two-stage greedy insertion procedure generates an initial solution s (Line 1). Then, a destroy operator O^- is selected from the set of destroy operators D based on the current scores SC (Line 4). Afterward, a repair operator O^+ is selected from the set of professional caregiver precedence based repair operators R_C if the number of non-improvement iterations t is no more than t_{max} (Lines 5–6) and from the set of general caregiver precedence based repair operators R_N (Lines 7–8) otherwise. The selected destroy and repair operators are successively performed on the current solution s to obtain the next solution s' (Line 10). Furthermore, a local search procedure is triggered to obtain the local optimum s'' (Line 11). If the objective value $f(s'')$ is better than $f(s)$, s''

is accepted as the new current solution. Meanwhile, the operators' scores SC are updated based on their performance at each iteration (Line 17). The ALNS algorithm is terminated when a maximum number of iterations $iter_{max}$ is reached. Finally, the post-optimization techniques are applied to derive solution s^* from the solution s produced by ALNS (Line 19). According to whether the objective value $f(s^*)$ is better than $f(s)$, s^{best} is updated by either s^* or s (Lines 20–24). The algorithm eventually returns the best solution s^{best} (Line 25). It's worth noting that the algorithm's ability to strike a balance between exploration and exploitation is facilitated by the various operators developed in both the destroy and repair phases, alongside an adaptive selection mechanism. The flow chart of the main scheme is depicted in Figure 3.

Algorithm 1 ALNS integrated with post-optimization techniques.

Input: a given problem instance NP
Output: best solution s^{best}

1: $s \leftarrow$ Two-StageGreedyInsertion(NP)
2: $t \leftarrow 0$, initialize the scores SC
3: **for** $iter = 1$ to $iter_{max}$ **do**
4:     $O^- \leftarrow$ SelectDestroyOperator(D, SC)
5:     **if** $t < t_{max}$ **then**
6:         $O^+ \leftarrow$ SelectRepairOperator(R_C, SC)
7:     **else**
8:         $O^+ \leftarrow$ SelectRepairOperator(R_N, SC)
9:     **end if**
10:     $s' \leftarrow$ PerformOperator(s, O^-, O^+)
11:     $s'' \leftarrow$ LocalSearch(s')
12:     **if** $f(s'') < f(s)$ **then**
13:         $s \leftarrow s''$
14:     **else**
15:         $t \leftarrow t{+}1$
16:     **end if**
17:     Update the scores SC
18: **end for**
19: $s^* \leftarrow$ Post-Optimize(s)
20: **if** $f(s^*) < f(s)$ **then**
21:     $s^{best} \leftarrow s^*$
22: **else**
23:     $s^{best} \leftarrow s$
24: **end if**
25: return s^{best}

4.2 Initial solution

The synchronization constraints are difficult to handle since any change in the scheduling of a single caregiver may affect the start time of the service. The classic greedy insertion (GI) algorithm ignores the interdependence among different routes, leading

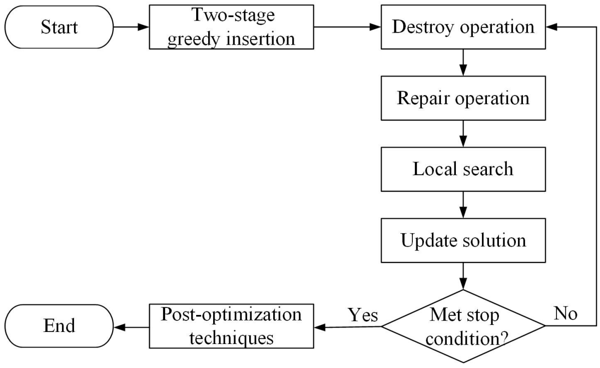

Fig. 3 The flow chat of main scheme

to infeasible solutions in preliminary tests. Therefore, we propose a two-stage collaborative scheduling strategy to address the synchronization constraints. The basic idea is that one of the caregivers starts the synchronized visit at a certain time, and the other caregiver strictly follows this time. Based on this idea, we propose a two-stage greedy insertion (TS-GI) procedure where the professional caregivers get precedence is developed to construct the initial solution. Algorithm 2 outlines the framework of the initial construction phase.

Algorithm 2 TS-GI structure.

Input: the set of patient's visits V; the set of caregivers K; related parameters of instance
Output: initial solution s

//First stage: schedule professional caregivers
for $k = 1$ to $|K|$ **do**
    Assign an empty route for each caregiver
    Determine the healthcare center for each professional caregiver to work
end for
Generate the set of all visits requiring professional caregivers $VisPoolPro \in V$
Determine the insertion order of visits in $VisPoolPro$
Initialize the set of synchronized visit's service time $ServiceTimeOfSynD$
for $i = 1$ to $|VisPoolPro|$ **do**
    Insert i at the end of such a route that minimizes the distance
    Update $ServiceTimeOfSynD$
end for
//Second stage: schedule general caregivers
Generate the set of general and synchronized visits $VisPoolGenSyn$ from V
Determine the insertion order of visits in $VisPoolGenSyn$
for $i = 1$ to $|VisPoolGenSyn|$ **do**
    Insert i at the best position of route that minimizes the variation in distance
end for
return s

In the first stage, each caregiver's route is initialized to be empty (Line 3). We sort the healthcare centers in non-increasing order by the total number of patients requiring type-II and type-III visits within their respective communities. Then, the professional caregivers are sequentially assigned to the ordered healthcare centers (Line 4), aiming to reduce the number of cross-community services. Following this, we start to build the professional caregivers' routes. We sort the patients requiring type-II and type-III visits in non-decreasing order of their time windows and use this as the insertion order (Lines 6–7). For each patient, we insert the required visit at the end of such a route that minimizes the distance variation (Line 10). Each time a patient is inserted, the service time for synchronized visits is updated (Line 11).

In the second stage, we consider the type-I visits by sorting them in non-decreasing order of the visit time windows. Meanwhile, we sort the type-III visits in non-decreasing order of their start service time, which was determined in the first stage. To preserve feasibility, we require that the insertion order of the type-I visits is ahead of the type-III visits (Lines 14–15). Next, we insert these visits into the general caregiver's routes with the lowest distance variation (Line 17). When a synchronized visit is inserted, the general caregiver cannot arrive after the service time determined in the first stage. If general caregivers arrive in advance, they need to wait for professional caregivers. We acknowledge that in some cases, general caregivers may need to work overtime to accommodate all visits while ensuring synchronization.

4.3 Destroy operators

Three classic destroy operators used in [25] are adopted to remove patients from the current solution and put them into the patient pool. These destroy operators are called Random Removal, Related Removal, and Worst Removal.

- `Random Removal`. This operator randomly selects q patients and removes them from the current routes.
- `Related Removal`. This operator randomly selects a patient from the solution as a seed patient and finds the $q-1$ patients closest to the seed patient. Remove the q patients from the current routes.
- `Worst Removal`. First, we calculates the removal gain of all the patients, defined as the difference between the objective with and without the patient in the solution. The q patients with the highest removal gains are removed from the current routes.

Based on the problem features of C-HHCRSP, we also design multiple new destroy operators that specially consider the synchronized visits for patients and the assignment of working healthcare centers for professional caregivers.

- `Synchronized Removal`. This operator randomly chooses q patients who require synchronized visits in the solution and removes them from the current routes.
- `Healthcare Center Switch`. This operator randomly selects m professional caregivers and assigns another healthcare center as the working place for each of them. Meanwhile, the patients assigned to these caregivers are removed and the traversed routes become empty.

- `Healthcare Center Close`. Out of all the healthcare centers where professional caregivers work, one of them is randomly selected to close. All the professional caregivers who work in this healthcare center, along with the assigned patients, are removed, and their routes become empty. Then, this operator chooses the healthcare center closest to the closed one as the working place for those removed professional caregivers.

4.4 Repair operators

4.4.1 Lunch break strategy

To evaluate the solution quality, we need to calculate the route cost of each caregiver in the solution. When the assigned patients and the order of route are determined for each caregiver, different lunch start times will have a major impact on the route cost. Since the lunch breaks should start before the caregivers begin serving a patient or after they finish some service, we need to set a visit in the route as the departure position before the lunch breaks start. Note that when the caregivers choose a community healthcare center as their departure position, they will start working after the lunch break. To determine the start time of lunch breaks, we propose two complementary strategies, namely the First Lunch Break (FLB) and the Best Lunch Break (BLB).

For the FLB strategy, if the time the caregivers return to their working center after finishing some visits is within the lunch break window, the first such visit is selected as the departure position before the lunch break. For the BLB strategy, the caregiver will select a visit position that yields the minimum route cost among all such visit positions. More precisely, for caregiver k working in the community healthcare center h^s, and the time he starts to serve patient i is ts_i^k. Let $ts_i^k + D_i + t_{ih^s}$ denote the time when the caregiver k returns to the healthcare center after finishing the visit i. When the condition $LL_k \leq ts_i^k + D_i + t_{ih^s} \leq LU_k - t_k^{Lun}$ stands, the visit position i satisfies the required time window of lunch break. Note that the FLB strategy can determine the start time of lunch breaks in a short time, while the BLB strategy ensures the best lunch break time.

4.4.2 Professional caregiver precedence based repair operators

The proposed ALNS algorithm employs two precedence-based repair operators for professional caregivers to reinsert removed patients into the destroyed solution: Professional-General Best Insertion and Professional-General Randomized Best Insertion.

- `Professional-General Best Insertion`. This operator adopts the same principle as TS-GI when dealing with synchronized visits. Therefore, the repair process performs the insertion of patients requiring type-II and type-III visits to the professional caregiver's route first and then the insertion of patients requiring type-I and type-III visits to the general caregiver's route. All the removed type-II and type-III visits are selected in random order and reinserted into such a feasible position that generates minimum insertion costs. The insertion cost is calculated by the objective variation. Each time such a visit is successfully inserted, the start service time of synchronized visits in the corresponding route is updated and regarded as their benchmark service time. In the next step, the removed type-I visits are considered randomly, while the type-III visits are sorted in non-decreasing order according to their start service time. We insert the type-I and type-III visits successively into the feasible position with the minimum insertion cost on the general caregivers' route. Once the synchronized visits are inserted, their benchmark time should be strictly followed. To determine the start time of caregivers' lunch breaks quickly, the FLB strategy is adopted each time a visit is inserted into the route, just as the strategy adopted by TS-GI.

- `Professional-General Randomized Best Insertion`. This operator is similar to Professional-General Best Insertion, except that the insertion cost is calculated by multiplying a random factor $p \in [0.8, 1.2]$ to increase randomness. Adding a perturbation mechanism helps the algorithm explore a broader solution space and improve the diversity of solutions.

Figure 4 illustrates the repair process of these two operators. The destroyed solution in Figure 4 requires the reinsertion of five type-I visits, one type-II visit, and one type-III visit into the route served by professional caregiver A, as well as general caregivers B and C. According to the two-stage collaborative strategy, professional caregivers should get precedence on synchronized visits. Thus, in the first stage, visits 12 and 14 are inserted into the feasible position on Caregiver A's route with minimum insertion costs. In the second stage, considering that general caregivers are limited to serving patients within their respective communities, visits 1, 2 and 5 are inserted into Caregiver B's route, while visits 6, 10 and 14 are inserted into Caregiver C's route. These insertions are also performed at feasible positions with minimum insertion costs.

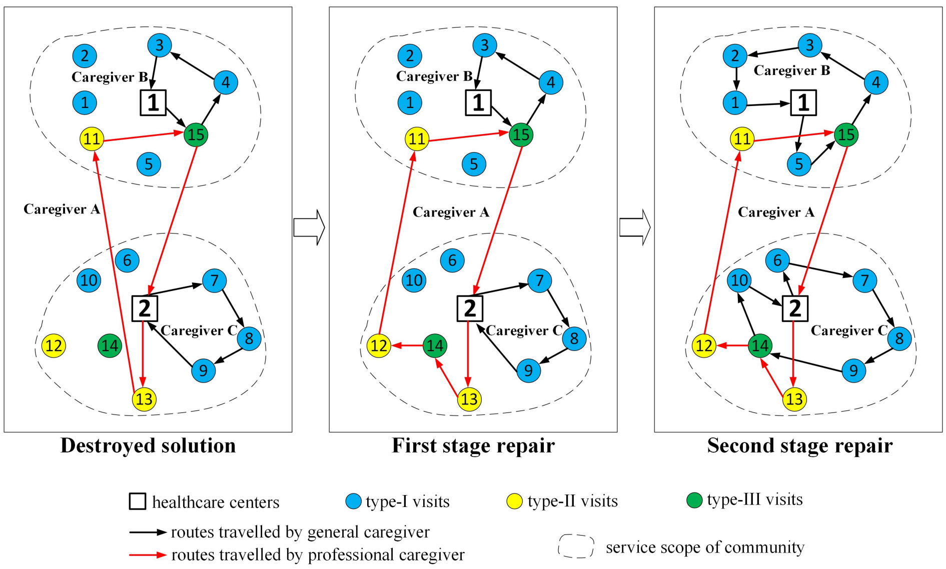

Fig. 4 Illustration of the professional caregiver precedence based repair operators

4.4.3 General caregiver precedence based repair operators

Recall that both the initial solution construction and the repair operators adopted the principle of first scheduling the professional caregivers when dealing with synchronization constraints, i.e., the start times of all the synchronized visits are always based on the arrival time of professional caregivers. If general caregivers arrive in advance, they need to wait. However, in some cases, long waiting times may lead to a situation in which the general caregivers cannot perform other visits after the synchronized visits, resulting in a high penalty cost for service delay. Thus, we propose two complementary repair operators that determine the start time of some synchronized visits based on general caregivers.

- `General-Professional Best Insertion`$^{\#1}$. The main difference with the Professional-General Best Insertion is that this operator inserts the patients who require type-I and type-III visits to the general caregiver's route first and then the patients who require type-II and type-III visits to the professional caregiver's route. This idea is beneficial to adjust the service time on certain synchronized visits to reduce the waiting time of general caregivers.
- `General-Professional Best Insertion`$^{\#2}$. This operator is derived from the General-Professional Best Insertion$^{\#1}$ by replacing the FLB strategy with the BLB strategy. While the FLB strategy can be used to compute the lunch times quickly, it may generate solutions of low quality. In contrast, the BLB strategy may find better lunch times for the caregivers but at the cost of extra computational efforts. With this operator, the algorithm can give a satisfying trade-off between computational time and solution quality.

4.5 Local search

A local search procedure is performed to optimize the solution obtained after performing a pair of destroy and repair operations. We adopt the swap(1,1) operator that exchanges two patients on the same routes, considering the difficulty of finding a feasible swap move that satisfies all the complex constraints of C-HHCRSP. In addition, the swap move operates only on two adjacent patients. This is based on the fact that a huge penalty cost of delays and increased waiting time of caregivers will occur if the time windows of the swapped patients are quite different.

4.6 Adaptive selection mechanism

The adaptive selection mechanism is designed to select the destroy/repair operators in each iteration efficiently. This mechanism embodies the principles of adaptivity and learning, where the performance of each operator influences its future selection probability. Initially, all operators are assigned with the same score SC_i. If a new global best solution is found by operator i, SC_i is increased by ξ. To implement this adaptive mechanism, the roulette wheel principle is used for operator selection. Each operator is metaphorically represented as a segment of a roulette wheel, with the size of the segment proportional to its score. As a result, the operator getting a higher score has a higher probability of being selected. The probability of selecting operator i, p_i, is calculated as $p_i = SC_i / \sum_{m=1}^{n} SC_m$, where n represents the total number of destroy/repair operators. In addition, it is crucial to adjust the precedence of different caregivers to improve the effectiveness of collaborative work on synchronized visits. Hence, the repair procedure begins by selecting an operator from the set of professional caregiver precedence based repair operators. After t_{max} iterations without improvement, the ALNS algorithm is supposed to be trapped in a local optimum. Then, we switch to use the general caregiver precedence based repair operators.

4.7 Post-optimization techniques

Upon the termination of the ALNS algorithm, two post-optimization techniques are applied to further enhance its performance: a heuristic strategy (HS) and a set partitioning (SP) model. The former involves a single-solution heuristic that refines the best solution derived from ALNS. Conversely, the latter amalgamates the high-quality solutions obtained during ALNS's search procedure to produce an improved solution.

4.7.1 Heuristic strategy

The proposed heuristic strategy, detailed in Algorithm 3, includes collaborative and workplace optimization. The collaborative optimization procedure is developed for synchronization constraints, which keeps the position of synchronized visits on the professional caregiver's route fixed and only adjusts their position on the current general caregiver's route. First, the patient who requires synchronized visits in $VisPooSyn$ is removed from the general caregiver's route, and the removal gain RG is calculated (Lines 4). When inserting the synchronized visits back, the start service time of these visits may change, impacting their time on the professional caregiver's route. Thus, the total insertion cost TC of the removed synchronized visits needs to consider both the insertion cost IC on the general caregiver's route and the cost change CC on the professional caregiver's route (Line 7). Finally, the removed synchronized visit is inserted into the feasible position with the lowest total insertion cost (Lines 8–13).

The workplace optimization aims to dispatch professional caregivers to the most appropriate healthcare center. This method keeps all professional caregiver's assigned patients and visiting orders fixed and only changes their workplace. For each professional caregiver in $ProK$, we traverse all the healthcare centers (Lines 15–17). Since changing the caregiver's workplace affects the start time of service, especially on synchronized visits, it is necessary to ensure that the current solution is still feasible on each route. The caregiver's workplace is updated when the cost of the resulting solution can be reduced (Lines 18–19).

4.7.2 Set partitioning model

In this section, we present the set partitioning based post-optimization technique. The idea is to maintain a set of high-quality routes found during the ALNS procedure and choose the best route for each caregiver by a set partitioning model to discover a better solution. Specifically, each time ALNS finds an improved solution s, we add it to the set S. Let c_s^k be the route cost of caregiver k in solution s. The parameter g_{is}^k

Algorithm 3 Heuristic strategy.

Create the set $VisPooSyn$ of synchronized visit and $ProK$ of professional caregiver
for $i = 1$ to $|VisPooSyn|$ **do**
  Find the professional route $RouP_i$ and general route $RouG_i$ which serve the patient i
  Remove i from route $RouG_i$ and calculate RG
  $TC \leftarrow 0$
  repeat
    Calculate IC on route $RouG_i$ and CC on route $RouP_i$
    if $RG+IC+CC < TC$ **then**
      $TC \leftarrow RG+IC+CC$
      Update the best insertion position
    end if
  until the end of route $RouG_i$
  Insert i at the best position of route $RouG_i$
end for
for $k = 1$ to $|ProK|$ **do**
  for All the healthcare center **do**
    Select the current healthcare center as workplace for caregiver k
    if Solution is feasible and its total cost is reduced **then**
      Update caregiver's workplace
    end if
  end for
end for

is assigned to 1 if caregiver k serves the patient i in solution s, and 0 otherwise. We use w_s^k to denote the binary decision variable, which equals 1 if the route of caregiver k in solution s is selected and 0 otherwise. Then, a problem-specific set partitioning model can be formulated as follows.

$$\min \ f = \sum_{k \in K} \sum_{s \in S} c_s^k w_s^k \tag{33}$$

$$\text{s.t.} \ \sum_{s \in S} w_s^k = 1 \quad \forall k \in K \tag{34}$$

$$\sum_{k \in K} \sum_{s \in S} g_{is}^k w_s^k = 1 \quad \forall i \in V_c \tag{35}$$

$$\sum_{k \in K_1} g_{is}^k w_s^k = \sum_{l \in K_2} g_{js}^l w_s^l \quad \forall (i,j) \in P^{syn}, s \in S \tag{36}$$

$$w_s^k \in \{0,1\} \quad \forall k \in K, s \in S \tag{37}$$

The objective (33) minimizes the total cost of routes, including the traveling cost, the penalty of delays and the caregivers' overtime. In other words, the objectives of the SP model and the model described in Section 3.2 are identical. Constraint (34) indicates that the route of each caregiver can only be selected from one solution. Constraint (35) ensures that each patient is served exactly once. Constraint (36) indicates that each synchronized visit must be served by a professional caregiver and a general caregiver simultaneously. Therefore, the routes of corresponding caregivers should be selected from the same solution. Constraint (37) defines the domains of the variable.

5 Computational results

The proposed algorithm is implemented in C++. The MIP and SP models are solved using CPLEX 12.9 with a single thread. All experiments are conducted on a server cluster with dual Intel® Xeon® Gold 6226R processors and 256 GB RAM on each node.

5.1 Instance generation

Since no benchmarks exist for C-HHCRSP, we generate instances by referring to the literature and real-life data. As C-HHCRSP can be regarded as a variant of the multi-depot heterogeneous vehicle routing problems with time windows, we refer to the setting of the locations of patients and depots, service duration and time windows from the literature. Specifically, patient locations are generated using both random uniform distribution (r) and a clustered-based method (c), following the methodologies outlined by Solomon [50], while healthcare center locations are set based on the works of Gillett and Johnson [20] and Cordeau et al [10]. We consider two types of time windows: narrow (n) and medium (m), with the width of narrow time windows being half of the corresponding medium time windows, as described by [11]. The center of each patient's time window is randomly selected from the interval $[50, 1200]$, and half of its width is generated as a normally distributed number. The lunch duration, lunchtime window, the proportion of synchronized visits and parameters relevant to the caregivers are generated by referring to Liu et al [31]. In particular, the lunchtime window is set to $[400, 600]$, and the lunch duration is 30. Considering the shortage of caregivers and the imbalance between supply and demand in real life, the number of caregivers is set to meet the demands or even necessitate some caregivers to work overtime to perform all the visits.

We generate nine groups of instances whose characteristics are summarized in Table 3. Column S represents the number of healthcare centers, and n denotes the number of patients situated in each center. N represents the total number of patients, N_1 denotes the number of patients who require type-II and type-III visits, and syn denotes the number of synchronized visits. k_1, k_2, and K denote the number of professional caregivers, general caregivers in each center and the total caregivers, respectively. t represents the type of time window, and L represents the location distribution. The other common parameters for the instances, as given in Table 4, are set based on intensive field research with a tertiary hospital in China. We generated 104 instances in total and denoted each one by InsS_n_syn_L_t. These instances can be downloaded from https://github.com/nwpu-orms/C-HHCRSP. We divide them into small, medium, large and huge instance sets according to the number of healthcare centers.

Table 3 Characteristics of each group

Size	Group	S	n	N	N_1	syn	k_1	k_2	K	t	L
small	1	2	6	12	3	1,2	1	1	3	n/m	r/c
	2	2	12	24	6	1,2,4	1	2	5	n/m	r/c
medium	3	4	12	48	12	3,5,9	2	2	10	n/m	r/c
	4	4	25	100	25	6,10,20	4	3	16	n/m	r/c
	5	4	50	200	50	12,20,40	8	6	32	n/m	r/c
large	6	6	25	150	36	9,15,30	6	3	24	n/m	r/c
	7	6	50	300	75	18,30,60	12	6	48	n/m	r/c
huge	8	8	25	200	50	12,20,40	8	3	32	n/m	r/c
	9	8	50	400	100	24,40,80	16	6	64	n/m	r/c

Table 4 The values of relevant parameters

Parameter	Description	Value
C^p	The unit penalty cost of delays for visits	10.00
C^g	The unit penalty cost of overtime for caregivers	10.00
α	The weight coefficient of route cost	0.40
β	The weight coefficient of penalty cost about delays	0.30
γ	The weight coefficient of penalty cost about overtime	0.30
ξ_1	The unit travel cost coefficient of professional caregivers	2.00
ξ_2	The unit travel cost coefficient of general caregivers	1.00

5.2 Parameter settings of ALNS

We employ the strategy proposed by Ropke and Pisinger [47] to determine the algorithm's parameters. Firstly, the initial values of all parameters are set through empirical observation. When tuning each parameter, we keep the values of other parameters unchanged. The algorithm was given five independent runs on small instances, and the parameter configuration with the best average results was selected. Table 5 displays the parameter values used in ALNS, where N and syn are instance-dependent parameters as given in Table 3.

Table 5 The values of relevant parameters in ALNS

Parameter	Description	Value
$iter_{max}$	Maximum number of iterations	30000
t_{max}	Maximum number of non-improvement iterations	10000
q^{Ran}	The number of removed patients in Random Removal	$0.2 \times N$
q^{Rel}	The number of removed patients in Related Removal	$0.3 \times N$
q^{Wor}	The number of removed patients in Worst Removal	$0.6 \times N$
q^{Syn}	The number of removed patients in Synchronized Removal	syn
SC_i	The initial score of each operator	100
ξ	Score increment in each iteration	10

5.3 Computational results and analysis

To evaluate the performance of the adaptive large neighborhood search algorithm integrated with the post-optimization techniques, this section conducts a comprehensive test based on the experimental instances designed in Section 5.1. Due to C-HHCRSP's NP-hardness, CPLEX can only solve the MIP model on small and medium instances. Therefore, we first analyze the performance of ALNS-HS by comparing it with the CPLEX solver. Then, we compare ALNS-HS and ALNS-SP with the classic LNS on larger instances. Furthermore, we discuss the effectiveness of the two post-optimization techniques, the scheduling of lunch breaks, and the proposed destroy/repair operators. Finally, a convergence analysis of ALNS and LNS is provided.

5.3.1 Comparison between ALNS-HS and CPLEX

We first compare ALNS-HS with CPLEX. The computational results are summarized in Table 6. For each instance, the detailed results of CPLEX and ALNS are presented. Columns *Obj* and *LB* give the best objective value and lower bound found by CPLEX within the time limit, respectively. If *Obj* equals *LB*, the solution is optimal, and the objective value is marked with an asterisk. Column *Time (s)* displays the computational time in seconds, and its maximum execution time is 3600 seconds. For ALNS-HS, columns *Best* and *Avg* present the best objective value and the average objective value found over five runs, respectively, and column *AvgTime (s)* shows the average running time of five runs. Column *Gap* presents the gap between *Best* and *Obj*, which is computed by $(Best - Obj)/\ Obj$. The best-found results are highlighted in boldface for each instance.

Table 6 shows that CPLEX can find optimal solutions quickly for small instances. As for medium instances, CPLEX cannot optimally solve all the instances with 4 healthcare centers and 12 patients per center within the given time limit. In contrast, ALNS-HS can obtain competitive solutions within a relatively short time. More specifically, CPLEX and ALNS-HS obtain the same objective values for 22 instances; for the remaining ten instances, CPLEX gives slightly better solutions for five instances, and ALNS-HS significantly outperforms CPLEX for the other five instances with gaps up to 12%.

5.3.2 Comparison between ALNS-HS, ALNS-SP and LNS

We compare the performance of our ALNS-HS and ALNS-SP algorithms with the classic LNS using the instances in Groups 4–9. The LNS algorithm was implemented using the destroy operators described in Section 4.3 and the repair operators described in Section 4.4.2, where each iteration sequentially selects the operators. In addition, the post-optimization process is not included in LNS.

Tables 7–9 report the computational results on medium, large and huge instances, respectively. These tables are organized similarly as Table 6. For convenience, we present the number of best solutions obtained by each algorithm in Figure 5. Both ALNS-HS and ALNS-SP significantly outperform LNS in terms of solution quality. For medium instances, ALNS-HS finds 21 better solutions, while ALNS-SP finds 19, with 16 ties. In contrast, LNS can not find even one better solution when compared

Table 6 Results for the small and some medium instances between CPLEX and ALNS-HS

Instance	CPLEX			ALNS-HS			Gap
	Obj	LB	Time (s)	Best	Avg	AvgTime (s)	
Ins2_6_1_c_m	**392.48***	392.48	0.10	**392.48***	392.48	3.38	-
Ins2_6_1_c_n	**161.88***	161.88	0.06	**161.88***	161.88	3.38	-
Ins2_6_1_r_m	**1194.52***	1194.52	0.16	**1194.52***	1194.52	3.05	-
Ins2_6_1_r_n	**1872.88***	1872.88	0.09	**1872.88***	1872.88	3.35	-
Ins2_6_2_c_m	**328.28***	328.28	0.10	**328.28***	328.28	3.38	-
Ins2_6_2_c_n	**188.73***	188.73	0.11	**188.73***	188.73	3.02	-
Ins2_6_2_r_m	**1132.05***	1132.05	0.10	**1132.05***	1132.05	3.25	-
Ins2_6_2_r_n	**1959.78***	1959.78	0.25	**1959.78***	1959.78	3.30	-
Ins2_12_1_c_m	**99.30***	99.30	0.72	**99.30***	99.30	11.39	-
Ins2_12_1_c_n	**1032.09***	1032.09	1.88	**1032.09***	1032.09	11.17	-
Ins2_12_1_r_m	**2167.54***	2167.54	3.12	2180.37	2180.37	9.16	0.59%
Ins2_12_1_r_n	**1359.67***	1359.67	0.97	**1359.67***	1359.67	10.53	-
Ins2_12_2_c_m	**649.07***	649.07	1.71	**649.07***	649.07	11.42	-
Ins2_12_2_c_n	**526.66***	526.66	1.01	**526.66***	526.66	10.30	-
Ins2_12_2_r_m	**1489.49***	1489.49	1.13	**1489.49***	1489.49	10.85	-
Ins2_12_2_r_n	**1985.09***	1985.09	3.13	2034.48	2034.48	10.86	2.49%
Ins2_12_4_c_m	**72.13***	72.13	3.04	**72.13***	72.13	13.64	-
Ins2_12_4_c_n	**722.17***	722.17	2.87	**722.17***	745.88	12.60	-
Ins2_12_4_r_m	**2284.72***	2284.72	12.12	**2284.72***	2284.72	15.05	-
Ins2_12_4_r_n	**2120.57***	2120.57	4.80	2172.54	2172.54	12.00	2.45%
Ins4_12_3_c_m	**1179.33***	1179.33	162.85	**1179.33***	1185.80	42.13	-
Ins4_12_3_c_n	**2153.34***	2153.34	1137.11	**2153.34***	2153.34	45.79	-
Ins4_12_3_r_m	**3801.50**	3534.97	3600.00	3875.02	3883.16	38.38	1.93%
Ins4_12_3_r_n	5337.10	4772.74	3600.00	**4811.61**	4811.61	43.61	-9.85%
Ins4_12_5_c_m	**502.45***	502.45	120.91	**502.45***	504.44	50.29	-
Ins4_12_5_c_n	**1073.79***	1073.79	711.39	1113.69	1113.69	45.04	3.72%
Ins4_12_5_r_m	3774.03	2882.71	3600.00	**3550.76**	3663.59	39.29	-5.92%
Ins4_12_5_r_n	5913.37	4149.52	3600.00	**5628.6**	5818.91	39.59	-4.82%
Ins4_12_9_c_m	**737.57**	711.17	3600.00	**737.57**	753.46	55.21	-
Ins4_12_9_c_n	**1004.84***	1004.84	770.48	**1004.84***	1057.07	50.64	-
Ins4_12_9_r_m	3691.13	2128.38	3600.00	**3447.67**	3585.40	45.21	-6.60%
Ins4_12_9_r_n	7217.50	4247.98	3600.00	**6336.40**	6336.40	46.57	-12.21%

with the two ALNS algorithms. For large instances, ALNS-HS and ALNS-SP find 17 and 19 out of 24 best solutions, with 13 ties. By comparison, LNS obtains one best solution. When solving huge instances, both ALNS-HS and ALNS-SP still outperform LNS. ALNS-HS finds 16 out of 24 best solutions, while ALNS-SP finds 18, with ten ties. However, no best solution can be attained by LNS.

Hence, we can conclude that both ALNS-HS and ALNS-SP are quite effective. ALNS-HS obtains better solutions than ALNS-SP on medium instances for most instances, while the latter performs better for large and huge instances. Furthermore, we conduct the statistical tests and prove that the performance of ALNS-HS is significantly better than the classic LNS, with the p-value of 1.42×10^{-5}, 1.25×10^{-5}, 9.71×10^{-6} for medium, large and huge instances, respectively. Similarly, ALNS-SP outperforms the classic LNS significantly, with the p-value of 1.44×10^{-5}, 1.1×10^{-5}, 2.34×10^{-5} for medium, large and huge instances.

Table 7 Results for the medium instances between ALNS-HS, ALNS-SP and LNS

Instance	ALNS-HS			ALNS-SP			LNS		
	Best	Avg	AvgTime (s)	Best	Avg	AvgTime (s)	Best	Avg	AvgTime (s)
Ins4_25_6_c_m	**852.97**	853.16	332.47	**852.97**	853.16	332.48	1059.47	1059.47	98.39
Ins4_25_6_c_n	**1559.31**	1565.13	281.06	**1559.31**	1565.13	281.06	1665.10	1666.20	107.20
Ins4_25_6_r_m	**4498.77**	4680.63	301.89	**4498.77**	4680.63	301.89	5073.87	5187.70	177.61
Ins4_25_6_r_n	6276.82	6421.49	201.09	**6087.16**	6331.41	201.09	6268.14	6374.47	100.87
Ins4_25_10_c_m	**345.57**	345.61	279.46	**345.57**	345.62	279.46	365.02	365.02	81.89
Ins4_25_10_c_n	**978.45**	1174.23	302.16	1175.58	1203.85	302.18	1254.19	1261.05	135.01
Ins4_25_10_r_m	**4496.90**	4517.14	296.70	4650.72	4692.14	296.71	4650.72	4666.12	127.74
Ins4_25_10_r_n	**8506.39**	8782.26	245.82	8526.71	8902.39	245.82	9067.78	9073.52	69.73
Ins4_25_20_c_m	**415.58**	421.90	308.49	**415.58**	422.07	308.48	422.11	426.90	84.59
Ins4_25_20_c_n	**1200.47**	1203.11	403.15	**1200.47**	1205.54	403.16	1290.54	1292.35	71.97
Ins4_25_20_r_m	**6778.87**	6888.51	333.91	**6778.87**	6901.21	333.91	6848.21	6921.59	76.13
Ins4_25_20_r_n	**6705.28**	6849.47	257.02	7081.26	7183.01	257.02	7245.42	7612.70	80.11
Ins4_50_12_c_m	**685.99**	688.88	2449.69	**685.99**	699.64	2449.71	697.70	701.63	902.30
Ins4_50_12_c_n	**1159.74**	1204.84	2142.12	**1159.74**	1237.26	2142.16	1298.39	1370.29	860.02
Ins4_50_12_r_m	**4354.93**	4454.23	2313.07	**4354.93**	4496.91	2313.09	4473.15	4627.48	1047.51
Ins4_50_12_r_n	**6828.17**	7013.43	1490.14	**6828.17**	6962.06	1490.18	7134.54	7213.83	578.18
Ins4_50_20_c_m	**667.88**	672.51	1683.49	**667.88**	672.25	1683.52	668.76	672.55	609.16
Ins4_50_20_c_n	**1191.45**	1199.44	2347.74	**1191.45**	1201.25	2347.79	1278.75	1280.44	956.94
Ins4_50_20_r_m	**4518.88**	4698.49	1835.06	**4518.88**	4685.02	1835.08	4778.28	4811.69	732.73
Ins4_50_20_r_n	**7927.66**	8151.41	1685.71	8039.97	8532.51	1685.73	8458.50	8522.10	728.20
Ins4_50_40_c_m	**886.35**	889.63	1732.17	**886.35**	891.74	1732.17	919.23	921.79	463.10
Ins4_50_40_c_n	864.26	914.88	1794.04	**848.09**	891.12	1794.07	904.00	956.32	481.10
Ins4_50_40_r_m	**6333.61**	6501.85	1502.65	**6333.61**	6535.24	1502.67	6506.91	6600.90	585.33
Ins4_50_40_r_n	7008.76	7224.48	1639.64	**6711.98**	7244.89	1639.67	7426.70	7701.83	341.28

Table 8 Results for the large instances between ALNS-HS, ALNS-SP and LNS

Instance	ALNS-HS			ALNS-SP			LNS		
	Best	Avg	AvgTime (s)	Best	Avg	AvgTime (s)	Best	Avg	AvgTime (s)
Ins6_25_9_c_m	**782.55**	808.79	607.45	**782.55**	806.08	607.46	860.35	871.98	191.24
Ins6_25_9_c_n	**1104.45**	1124.69	823.73	**1104.45**	1122.80	823.75	1143.53	1150.93	412.69
Ins6_25_9_r_m	**5040.98**	5110.77	517.12	**5040.98**	5052.07	517.11	5059.50	5163.91	366.90
Ins6_25_9_r_n	**7689.16**	7742.94	485.53	**7689.16**	7768.31	485.54	7959.59	7968.57	181.77
Ins6_25_15_c_m	**981.66**	1027.19	596.32	**981.66**	1024.30	596.34	1064.67	1093.19	196.10
Ins6_25_15_c_n	**1466.30**	1489.56	621.02	**1466.30**	1499.56	621.04	1694.71	1776.68	137.21
Ins6_25_15_r_m	**3900.87**	3999.04	581.89	3914.31	4077.54	581.90	4155.46	4233.01	181.36
Ins6_25_15_r_n	**9529.90**	9613.25	514.68	**9529.90**	9636.39	514.68	9898.53	9916.28	186.52
Ins6_25_30_c_m	621.67	649.17	565.67	619.75	644.07	565.66	**618.77**	643.79	154.38
Ins6_25_30_c_n	**2560.91**	2673.63	564.25	2668.36	2748.64	564.26	2699.64	2745.81	127.51
Ins6_25_30_r_m	**8095.01**	8204.39	576.70	8604.16	8728.27	576.69	8964.72	9053.76	138.38
Ins6_25_30_r_n	9778.31	9919.19	667.38	**9554.42**	9986.19	667.38	9784.15	9866.50	187.93
Ins6_50_18_c_m	**1013.31**	1023.87	4998.68	**1013.31**	1023.20	4998.71	1037.53	1042.58	1721.49
Ins6_50_18_c_n	1771.97	1787.63	4951.15	**1759.22**	1783.97	4951.14	1772.81	1792.49	2173.96
Ins6_50_18_r_m	6479.57	6576.93	5406.49	**6471.75**	6530.94	5406.54	6570.21	6660.41	2772.48
Ins6_50_18_r_n	**7953.03**	8257.60	4429.85	8091.80	8276.66	4429.86	8297.96	8450.78	2693.39
Ins6_50_30_c_m	**1172.70**	1190.12	5084.21	**1172.70**	1182.4	5084.28	1179.42	1185.19	1969.33
Ins6_50_30_c_n	**1427.01**	1480.42	6572.88	**1427.01**	1480.49	6572.94	1501.95	1526.74	1981.56
Ins6_50_30_r_m	6312.91	6461.87	4194.67	**6252.07**	6419.62	4194.73	6586.57	6684.85	1874.37
Ins6_50_30_r_n	8256.48	8571.94	5824.10	**8041.70**	8394.96	5824.16	8653.80	8796.69	1888.66
Ins6_50_60_c_m	**959.24**	975.45	6134.07	**959.24**	971.65	6134.08	996.89	1019.08	886.48
Ins6_50_60_c_n	**1674.61**	1713.07	5265.80	**1674.61**	1707.95	5265.79	1760.15	1765.69	1084.32
Ins6_50_60_r_m	6319.30	6476.10	5450.85	**6293.41**	6544.30	5450.82	6576.74	6714.84	984.91
Ins6_50_60_r_n	**9616.88**	9887.17	6382.36	**9616.88**	10091.72	6382.36	9943.98	10062.12	1227.40

Table 9 Results for the huge instances between ALNS-HS, ALNS-SP and LNS

Instance	ALNS-HS			ALNS-SP			LNS		
	Best	Avg	AvgTime (s)	Best	Avg	AvgTime (s)	Best	Avg	AvgTime (s)
Ins8_25_12_c_m	**996.61**	1017.75	1181.32	**996.61**	1036.85	1181.32	1020.65	1027.46	793.51
Ins8_25_12_c_n	**2495.28**	2523.37	989.15	**2495.28**	2506.01	989.16	2587.05	2636.56	421.17
Ins8_25_12_r_m	7697.02	7814.83	1196.23	**7690.27**	7773.21	1196.23	7861.59	7904.76	426.79
Ins8_25_12_r_n	**10941.20**	11186.86	1055.49	10993.20	11296.22	1055.49	11222.80	11352.78	638.03
Ins8_25_20_c_m	1182.63	1239.28	1162.53	**1168.26**	1189.45	1162.55	1253.40	1286.83	676.79
Ins8_25_20_c_n	**2879.44**	2917.25	1207.54	2885.63	2903.98	1207.56	3003.94	3045.42	341.34
Ins8_25_20_r_m	8141.83	8348.92	1191.42	**8022.45**	8164.01	1191.44	8465.46	8556.75	356.04
Ins8_25_20_r_n	**11755**	11992.76	1041.04	11802.20	12089.60	1041.04	12456.00	12585.92	387.46
Ins8_25_40_c_m	**925.09**	950.47	1004.56	929.10	971.09	1004.55	982.00	1016.51	328.48
Ins8_25_40_c_n	**2091.80**	2111.20	1187.93	2254.25	2278.62	1187.91	2274.07	2333.75	269.49
Ins8_25_40_r_m	**7537.37**	7711.20	1400.01	**7537.37**	7555.81	1400.02	7695.48	7847.73	320.12
Ins8_25_40_r_n	13619.70	13706.64	1448.67	**13491.50**	13770.92	1448.68	13799.30	13938.74	355.09
Ins8_50_24_c_m	**1414.42**	1437.56	11761.84	**1414.42**	1440.69	11761.90	1450.81	1464.19	6546.20
Ins8_50_24_c_n	2351.01	2408.83	14589.84	**2260.53**	2331.10	14589.96	2356.35	2400.78	7501.67
Ins8_50_24_r_m	8309.26	8583.76	10605.18	**8298.78**	8551.19	10605.20	8767.07	8824.91	5175.61
Ins8_50_24_r_n	11476.60	11854.12	11920.90	**11191.40**	11660.24	11920.88	11892.70	12081.22	4277.20
Ins8_50_40_c_m	**1255.79**	1276.27	11706.42	**1255.79**	1280.09	11706.40	1268.60	1276.97	5081.43
Ins8_50_40_c_n	**2581.30**	2618.04	13093.18	**2581.30**	2613.51	13093.18	2751.34	2809.89	3621.31
Ins8_50_40_r_m	**8186.54**	8406.46	13956.30	**8186.54**	8458.48	13956.32	8269.78	8370.02	4196.68
Ins8_50_40_r_n	**14938.00**	15178.56	15006.16	**14938.00**	15643.90	15006.12	15578.00	15929.68	4981.32
Ins8_50_80_c_m	1254.51	1278.72	13447.34	**1246.07**	1277.51	13447.14	1264.88	1301.37	2428.40
Ins8_50_80_c_n	**3531.30**	3626.55	15626.34	**3531.30**	3611.34	15626.26	3691.41	3719.57	3703.70
Ins8_50_80_r_m	**10154.60**	10433.88	17346.28	**10154.60**	10416.46	17346.16	10653.60	10853.42	2828.15
Ins8_50_80_r_n	**16193.80**	16792.12	15336.32	16378.30	17926.16	15336.10	17035.10	17368.34	2534.53

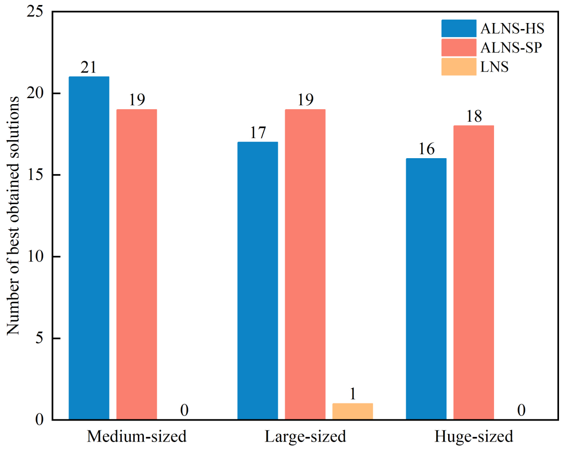

Fig. 5 Number of best solutions obtained by three algorithms

5.3.3 Effectiveness analysis of post-optimization techniques

To assess whether the post-optimization techniques can help ALNS perform better, we record the improvement of solutions when implementing such techniques. Table 10 shows the average improvement of objective values and the number of instances improved by the HS and SP based post-optimization techniques. We observe that both HS and SP can effectively improve the solution quality. HS is able to obtain improved solutions for 3, 7, 4 and 7 instances over different sizes. SP is able to find improved solutions for 0, 4, 10 and 8 instances over different sizes. As the size of the instance increases, the objective improvement of the HS based post-optimization technique gradually decreases, while the counterpart SP based post-optimization technique remains relatively stable.

Table 10 Effective of HS method and SP model

Size	Number of instances		Improvement of object value	
	HS	SP	HS	SP
Small	3/20	0/20	3.36%	0.00%
Medium	7/36	4/36	1.16%	0.33%
large	4/24	10/24	0.65%	0.46%
huge	7/24	8/24	0.53%	0.46%

The computation times of the post-optimization techniques on each group of instances are visualized in Figure 6. As seen from Figure 6, both the HS and SP are highly effective and can further optimize ALNS within an extremely short time.

Computation times increase as the total number of patients grows. Moreover, the HS-based post-optimization technique is faster for instances in groups 1 to 8 but slower in group 9 than its counterpart SP.

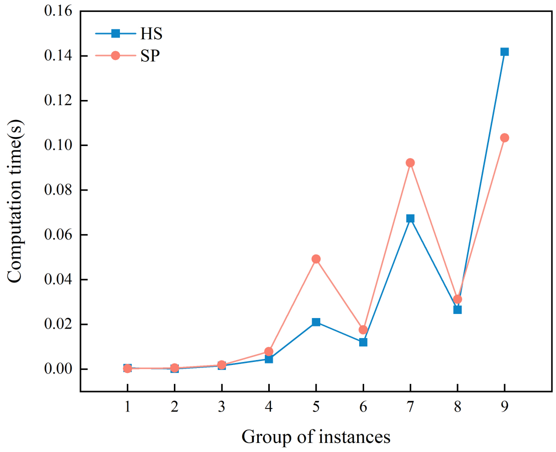

Fig. 6 Computation times of post-optimization techniques on each group of instances

5.3.4 Effectiveness analysis of lunch break scheduling

The time when caregivers should take a lunch break needs to be determined in our proposed model. To evaluate the impact, we proposed a modified model where the lunch break time is fixed. The difference between the modified model and the original model lies in the replacement of constraints (22)-(25) with constraints (38)-(41) to fix the lunch break time TL of each caregiver. We set TL at 400, 500, and 600, respectively, to test the modified model's performance under different fixed lunch times. We solve this modified model by the CPLEX solver on small instances in the same experimental environment.

$$ts_i^k + D_i + t_{ih^s} \geq TL + M(\sum_{j \in V_C^h \cup h} z_{ij}^k - 1) \quad \forall h \in V_D, k \in K_2^h, i \in V_C^h \cup h \tag{38}$$

$$ts_i^k + D_i + t_{ih^s} \leq TL + M(1 - \sum_{j \in V_C^h \cup h} z_{ij}^k) \quad \forall h \in V_D, k \in K_2^h, i \in V_C^h \cup h \tag{39}$$

$$ts_i^k + D_i + t_{ih^s} \geq TL + M(y_{h^f}^k + \sum_{j \in V_C \cup h} z_{ij}^k - 2) \quad \forall h \in V_D, k \in K_1, i \in V_C \cup h \tag{40}$$

$$ts_i^k + D_i + t_{ih^s} \leq TL + M(2 - y_{h^f}^k - \sum_{j \in V_C \cup h} z_{ij}^k) \quad \forall h \in V_D, k \in K_1, i \in V_C \cup h \tag{41}$$

Table 11 reports the computational results. We observe that the original model with a scheduled lunch break gets a better objective value than the modified model with fixed lunchtime ($TL = 400$, $TL = 500$, $TL = 600$) for all the tested instances. By analyzing the solutions, we find that when lunch breaks are fixed, caregivers have a higher probability of delaying service or working overtime, resulting in additional penalty costs. Moreover, the computational time required to solve the modified model has also increased significantly. These results indicate the importance of scheduling lunch break time for caregivers.

Table 11 Results for the proposed model with scheduled lunch break and fixed lunch time

Instance	Scheduled lunch break		$TL = 400$		$TL = 500$		$TL = 600$	
	Obj	Time (s)	Obj	Time (s)	Obj	Time (s)	Obj	Time (s)
Ins2_6_1_c_m	**392.48**	0.10	531.16	0.07	816.19	0.14	1568.45	0.31
Ins2_6_1_c_n	**161.88**	0.06	1345.02	0.08	908.25	0.07	234.17	0.07
Ins2_6_1_r_m	**1194.52**	0.16	1735.39	0.19	3483.98	2.69	1854.19	0.37
Ins2_6_1_r_n	**1872.88**	0.09	2392.27	0.15	2324.48	0.18	2924.48	0.19
Ins2_6_2_c_m	**328.28**	0.10	**328.28**	0.08	719.47	0.21	567.79	0.17
Ins2_6_2_c_n	**188.73**	0.11	1201.97	0.17	1169.04	0.43	1460.64	0.70
Ins2_6_2_r_m	**1132.05**	0.10	**1132.05**	0.10	1971.61	0.18	2538.03	0.18
Ins2_6_2_r_n	**1959.78**	0.25	3129.67	1.54	3369.30	2.48	4012.44	2.63
Ins2_12_1_c_m	**99.30**	0.72	209.59	1.10	1005.86	6.70	1668.99	9.60
Ins2_12_1_c_n	**1032.09**	1.88	1882.66	2.48	1991.37	4.96	2547.92	2.92
Ins2_12_1_r_m	**2167.54**	3.12	3284.73	9.97	3393.86	10.87	2927.91	8.25
Ins2_12_1_r_n	**1359.67**	0.97	2531.95	3.53	2669.04	7.66	2588.31	5.53
Ins2_12_2_c_m	**649.07**	1.71	931.21	2.46	2345.40	6.02	851.69	1.67
Ins2_12_2_c_n	**526.66**	1.01	1353.11	4.46	1787.85	4.50	2061.11	12.22
Ins2_12_2_r_m	**1489.49**	1.13	3368.40	5.41	2597.18	3.74	3332.12	10.55
Ins2_12_2_r_n	**1985.09**	3.13	3533.48	15.55	2589.03	4.42	3388.37	15.71
Ins2_12_4_c_m	**72.13**	3.04	1036.84	5.73	419.97	3.58	928.43	6.47
Ins2_12_4_c_n	**722.17**	2.87	1185.10	2.30	1715.97	4.15	1725.10	2.91
Ins2_12_4_r_m	**2284.72**	12.12	3494.17	46.69	3630.03	41.40	3518.54	41.82
Ins2_12_4_r_n	**2120.57**	4.80	2789.96	7.85	2730.54	10.91	5480.64	259.55

5.3.5 Effectiveness analysis of destroy/repair operators

To evaluate the impact of the destroy/repair operators, we conduct the following experiments. The first experiment disables one operator from the ALNS algorithm and reports the objective value deviations. The second experiment summarizes the average number of improved solutions found by each operator. Specifically, each time an improved solution is found by performing a pair of destroy and repair operators, the improvement count of the two operators is increased by one.

Table 12 displays computational statistics of each operator over five runs. In the set of destroy operators, Random Removal is found to be the most effective by obtaining an objective value deviation of 3.49% and 38.7 improved solutions on average. Moreover, our three new operators, including Synchronized Removal, Healthcare Center Switch and Healthcare Center Close, demonstrate their positive roles in the performance of the ALNS algorithm. As for the repair operators, the operators based on the professional caregiver precedence show complementary results to those based on general caregiver precedence. The former finds more improved solutions, while the latter has more impact on solution quality. This indicates that when dealing with synchronization constraints, the best strategy is not to always schedule professional caregivers first for all synchronized visits but rather to coordinate the working hours of professional and general caregivers. It is worth noting that even though the General-Professional Best Insertion$^{\#2}$ finds improved solutions less frequently than the Professional-General Best Insertion, the impact of the former on the solution quality is more significant. This is due to the fact that the BLB strategy will determine the optimal start time for lunch breaks, resulting in a substantial improvement in the solution quality within a relatively small number of iterations. To sum up, each of the ten operators shows its merit in the algorithm performance.

Table 12 Computational statistics of each operator

Operator	Objective value deviation without this operator	Average number of improved solutions found by the operator
Destroy operator:		
Random Removal	3.49%	38.70
Related Removal	0.67%	4.96
Worst Removal	0.44%	0.68
Synchronized Removal	0.81%	5.81
Healthcare Center Switch	1.14%	3.75
Healthcare Center Close	0.80%	1.94
Repair operator:		
Professional-General Best Insertion	0.46%	31.65
Professional-General Randomized Best Insertion	0.55%	21.20
General-Professional Best Insertion$^{\#1}$	0.65%	0.53
General-Professional Best Insertion$^{\#2}$	3.40%	2.46

5.3.6 Convergence analysis of ALNS and LNS

The convergence chart in Figure 7 illustrates the variation of objective values with increasing iterations for the instance denoted by Ins4_25_10_r_n on ALNS and LNS algorithms. The horizontal axis represents the number of iterations, while the vertical axis indicates the gap between the objective value obtained at the current iteration and the best-found objective value. We can observe that within the initial 500 iterations, the gaps notably decrease from 240% to 5% for ALNS and from 240% to 10% for LNS. However, as the iterations progress, the gaps decrease much slower and eventually converge toward optimal or suboptimal solutions. Note that a similar trend can be observed in other instances. The convergence results confirm the superiority of our proposed ALNS algorithm.

6 Conclusion

This paper investigated a new collaborative home health care routing and scheduling problem in multiple service centers. Many realistic complex constraints are considered, including time windows, mandatory lunch breaks, synchronized visits, downgrading services, etc. A mixed integer programming model is constructed, and the general CPLEX solver can solve small instances. To solve larger instances, we additionally proposed an adaptive large neighborhood search (ALNS) algorithm by combining three classic and seven destroy/repair operators specially designed for the problem. The adaptive selection mechanism of the destroy/repair operators significantly enhanced the collaborative effect among caregivers. Moreover, we proposed post-optimization techniques based on heuristic strategies and a set partitioning model at the end of

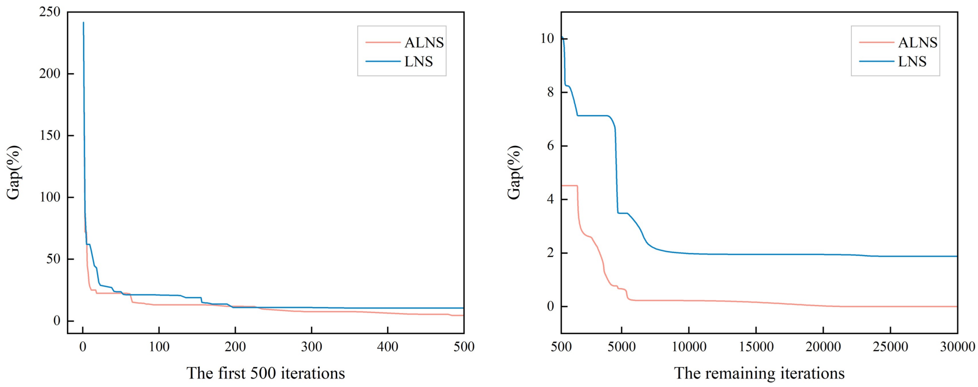

Fig. 7 The convergence process of instance Ins4_25_10_r_n

ALNS for performance enhancement. Experimental tests on 104 instances demonstrated that our proposed ALNS algorithm competes well with the CPLEX solver on 32 small instances. For instances where CPLEX fails to produce feasible solutions, both ALNS-based algorithms significantly outperform the classic large neighborhood search algorithm. The role of the destroy/repair operators in the performance of the algorithm is shown by additional experimental analysis.

In future work, we plan to focus on uncertain aspects of home health care in practice, such as stochastic traveling times of caregivers and service duration uncertainty. As these stochastic factors affect the quality of services, we intend to design stochastic programming methods to deal with them. We also plan to extend the planning horizon of this problem from one day to several days and to introduce patient preferences to improve service levels.

References

[1] Ait Haddadene SR, Labadie N, Prodhon C (2019) Bicriteria vehicle routing problem with preferences and timing constraints in home health care services. Algorithms 12(8):152

[2] Akbari V, Sadati İ, Salman FS, et al (2023) Minimizing total weighted latency in home healthcare routing and scheduling with patient prioritization. Or Spectrum pp 1–46

[3] Bachouch RB, Guinet A, Hajri-Gabouj S (2011) A decision-making tool for home health care nurses' planning. Supply Chain Forum: An International Journal 12(1):14–20

[4] Bard JF, Shao Y, Jarrah AI (2014) A sequential grasp for the therapist routing and scheduling problem. Journal of Scheduling 17:109–133

[5] Barrera D, Velasco N, Amaya CA (2012) A network-based approach to the multi-activity combined timetabling and crew scheduling problem: Workforce scheduling for public health policy implementation. Computers & Industrial Engineering 63(4):802–812

[6] Bredström D, Rönnqvist M (2008) Combined vehicle routing and scheduling with temporal precedence and synchronization constraints. European Journal of Operational Research 191(1):19–31

[7] Cappanera P, Scutellà MG (2015) Joint assignment, scheduling, and routing models to home care optimization: A pattern-based approach. Transportation Science 49(4):830–852

[8] Carello G, Lanzarone E (2014) A cardinality-constrained robust model for the assignment problem in home care services. European Journal of Operational Research 236(2):748–762

[9] Chen H, Luo X, Zhang Z, et al (2021) Stochastic bi-level programming model for home healthcare scheduling problems considering the degree of satisfaction with visit time. Journal of Systems Science and Systems Engineering 30:572–599

[10] Cordeau JF, Gendreau M, Laporte G (1997) A tabu search heuristic for periodic and multi-depot vehicle routing problems. Networks: An International Journal 30(2):105–119

[11] Cordeau JF, Laporte G, Mercier A (2001) A unified tabu search heuristic for vehicle routing problems with time windows. Journal of the Operational Research Society 52(8):928–936

[12] Decerle J, Grunder O, El Hassani AH, et al (2018) A memetic algorithm for a home health care routing and scheduling problem. Operations research for health care 16:59–71

[13] Dohn A, Kolind E, Clausen J (2009) The manpower allocation problem with time windows and job-teaming constraints: A branch-and-price approach. Computers & Operations Research 36(4):1145–1157

[14] Erdem M, Koç Ç (2023) Home health care and dialysis routing with electric vehicles and private and public charging stations. Transportation Letters 15(5):423–438

[15] Fathollahi-Fard AM, Hajiaghaei-Keshteli M, Mirjalili S (2020) A set of efficient heuristics for a home healthcare problem. Neural Computing and Applications 32:6185–6205

[16] Fernandez A, Gregory G, Hindle A, et al (1974) A model for community nursing in a rural county. Journal of the Operational Research Society 25(2):231–239

[17] Fikar C, Hirsch P (2015) A matheuristic for routing real-world home service transport systems facilitating walking. Journal of Cleaner Production 105:300–310

[18] Frifita S, Masmoudi M (2020) Vns methods for home care routing and scheduling problem with temporal dependencies, and multiple structures and specialties. International Transactions in Operational Research 27(1):291–313

[19] Fu Y, Ma X, Gao K, et al (2023) Multi-objective home health care routing and scheduling with sharing service via a problem-specific knowledge-based artificial bee colony algorithm. IEEE Transactions on Intelligent Transportation Systems

[20] Gillett BE, Johnson JG (1976) Multi-terminal vehicle-dispatch algorithm. Omega 4(6):711–718

[21] Grenouilleau F, Legrain A, Lahrichi N, et al (2019) A set partitioning heuristic for the home health care routing and scheduling problem. European Journal of Operational Research 275(1):295–303

[22] Guericke D, Suhl L (2017) The home health care problem with working regulations. Or Spectrum 39:977–1010

[23] Hashemi Doulabi H, Pesant G, Rousseau LM (2020) Vehicle routing problems with synchronized visits and stochastic travel and service times: Applications in healthcare. Transportation Science 54(4):1053–1072

[24] Heching A, Hooker JN, Kimura R (2019) A logic-based benders approach to home healthcare delivery. Transportation Science 53(2):510–522

[25] Hemmelmayr VC, Cordeau JF, Crainic TG (2012) An adaptive large neighborhood search heuristic for two-echelon vehicle routing problems arising in city logistics. Computers & Operations Research 39(12):3215–3228

[26] Hertz A, Lahrichi N (2009) A patient assignment algorithm for home care services. Journal of the Operational Research Society 60(4):481–495

[27] Hiermann G, Prandtstetter M, Rendl A, et al (2015) Metaheuristics for solving a multimodal home-healthcare scheduling problem. Central European Journal of Operations Research 23:89–113

[28] Holm SG, Angelsen RO (2014) A descriptive retrospective study of time consumption in home care services: how do employees use their working time? BMC Health Services Research 14(1):1–10

[29] Li H, Wang H, Chen J, et al (2021) Two-echelon vehicle routing problem with satellite bi-synchronization. European Journal of Operational Research 288(3):775–793

[30] 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

[31] Liu W, Dridi M, Fei H, et al (2021) Hybrid metaheuristics for solving a home health care routing and scheduling problem with time windows, synchronized visits and lunch breaks. Expert Systems with Applications 183:115307

[32] Mankowska DS, Meisel F, Bierwirth C (2014) The home health care routing and scheduling problem with interdependent services. Health Care Management Science 17:15–30

[33] Masmoudi M, Jarboui B, Borchani R (2023) Efficient metaheuristics for the home (health)-care routing and scheduling problem with time windows and synchronized visits. Optimization Letters pp 1–33

[34] Méndez-Fernández I, Lorenzo-Freire S, González-Rueda ÁM (2023) An adaptive large neighbourhood search algorithm for a real-world home care scheduling problem with time windows and dynamic breaks. Computers & Operations Research 159:106351

[35] Mendoza-Alonzo J, Zayas-Castro J, Charkhgard H (2020) Office-based and home-care for older adults in primary care: A comparative analysis using the nash bargaining solution. Socio-Economic Planning Sciences 69:100710

[36] Milburn AB, Spicer J (2013) Multi-objective home health nurse routing with remote monitoring devices. International Journal of Planning and Scheduling 1(4):242–263

[37] Mısır M, Smet P, Vanden Berghe G (2015) An analysis of generalised heuristics for vehicle routing and personnel rostering problems. Journal of the Operational Research Society 66(5):858–870

[38] Mutingi M, Mbohwa C (2014) Multi-objective homecare worker scheduling: A fuzzy simulated evolution algorithm approach. IIE Transactions on Healthcare Systems Engineering 4(4):209–216

[39] Naderi B, Begen MA, Zaric GS, et al (2023) A novel and efficient exact technique for integrated staffing, assignment, routing, and scheduling of home care services under uncertainty. Omega 116:102805

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

[41] Nikzad E, Bashiri M, Abbasi B (2021) A matheuristic algorithm for stochastic home health care planning. European Journal of Operational Research

288(3):753–774

[42] Rasmussen MS, Justesen T, Dohn A, et al (2012) The home care crew scheduling problem: Preference-based visit clustering and temporal dependencies. European Journal of Operational Research 219(3):598–610

[43] 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

[44] Rest KD, Hirsch P (2016) Daily scheduling of home health care services using time-dependent public transport. Flexible Services and Manufacturing Journal 28:495–525

[45] Riazi S, Wigström O, Bengtsson K, et al (2018) A column generation-based gossip algorithm for home healthcare routing and scheduling problems. IEEE Transactions on Automation Science and Engineering 16(1):127–137

[46] Rodriguez C, Garaix T, Xie X, et al (2015) Staff dimensioning in homecare services with uncertain demands. International Journal of Production Research 53(24):7396–7410

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

[48] Shao Y, Bard JF, Jarrah AI (2012) The therapist routing and scheduling problem. IIE Transactions 44(10):868–893

[49] Soares R, Marques A, Amorim P, et al (2023) Synchronisation in vehicle routing: classification schema, modelling framework and literature review. European Journal of Operational Research

[50] Solomon M (1987) Algorithms for the vehicle routing and scheduling problems with time window constraints. Operations Research 35:254–265

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

[52] Yazır OA, Koç Ç, Yücel E (2023) The multi-period home healthcare routing and scheduling problem with electric vehicles. Or Spectrum pp 1–49

[53] Yin Y, Liu X, Chu F, et al (2023) An exact algorithm for the home health care routing and scheduling with electric vehicles and synergistic-transport mode. Annals of Operations Research pp 1–36

[54] 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