Assignment and Routing in the Home Health Care Problem*

Shihhung Su[1][0009-0005-4961-8746] and Chiuyuan Chen[2][0000-0002-3716-3810]

[1] National Yang Ming Chiao Tung University, Taiwan
shihong253@gmail.com
[2] National Yang Ming Chiao Tung University, Taiwan
cychen@nycu.edu.tw

Abstract. This study addresses the Home Health Care (HHC) problem with the goals of minimizing both the cost of hiring caregivers and the transportation cost incurred as caregivers travel between patients' geographic coordinates of residences, while satisfying patient-friendliness constraints. To achieve the aforementioned goals, this study proposes a new caregiver assignment and routing algorithm, called hiCH. Our algorithm is a heuristic algorithm. It does not require a set of caregivers to be given in advance, but instead hires caregivers one by one as the problem is addressed, thereby reducing hiring costs. Our algorithm hiCH adopts a hierarchical convex hull approach, considering the convex hull formed by unvisited patients and selecting a patient with the highest care level required on the convex hull to determine the employment level of the caregiver and serve as the starting point of the caregiver's visiting path, with the aim of reducing transportation cost. We improve hiCH by proposing D-hiCH, in which patients requests are further sorted according to their required service levels and time windows. To further improve solution quality, we also propose a genetic algorithm (GA) version of D-hiCH.

Keywords: Timetabling in Home Health Care (HHC) · assignment · routing · convex hull · genetic algorithm (GA).

1 Introduction

The aging population has become a pressing global concern, leading to a rapidly growing demand for medical services, particularly among individuals with chronic illnesses or limited mobility. In response to this challenge, Home Health Care (HHC) has emerged as a vital solution. HHC promotes care that is not only more accessible but often more cost-effective and better aligned with long-term health outcomes. See references [3] and [6].

The two primary components in an HHC system are: patients and caregivers. Patients submit requests for services that detail their specific care requirements

* This research was partially supported by the National Science and Technology Council of the Republic of China under the grant NSTC 113-2115-M-A49-012-MY2.

(or demands), such as the duration of the required service, the time window available for the assigned caregiver to visit, and the needed skill level of the assigned caregiver. The HHC problem focus on assigning all requests to sufficient number of caregivers and constructing feasible schedules for the caregivers that respect working time regulations, while at the same time, trying to minimize the cost caused by hiring caregivers and transportation between geographic coordinates of residences (or simply residences) of patients.

Based on the above, the two major decision-making aspects of the HHC problem are: assignment and routing. Assignment involves determining the number of caregivers required and allocating each patient's request to a suitable caregiver. Routing involves determining the sequence in which each caregiver visits the assigned patients. When modeling the above two major decision-making aspects, a distinction arises regarding how a caregiver's working time is defined.

Let $(*)$ denote the depot or the caregivers' residence. We say that a caregiver's working time is of *type 1* (respectively, *type 2*) if the travelling time from $(*)$ to the first patient and from the last patient back to $(*)$ is included (respectively, excluded) in the working time, as shown in Figure 1.

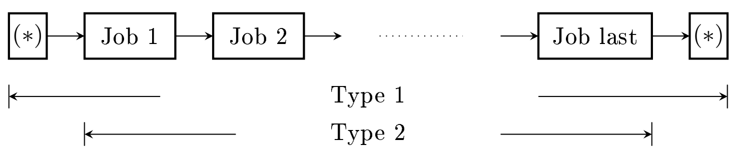


Fig. 1. Two types of working time: $(*)$ means the depot or the caregiver's residence.

The most related works of this manuscript are [1, 3, 5, 6]. We summarize these related works in Table 1 and briefly describe these works below.

Reference [3] considers weekly planning of HHC; and, the HHC problem is decomposed into four sequential sub-problems: (1) districting, (2) dimensioning, (3) assignment, and (4) routing. These four sub-problems are integrated and formulated as a mixed-integer optimization problem. In [3], all caregivers' routes must start and end at their residences and the working time is of type 1. The districting step significantly reduces the problem size, leading to notable improvements in computational efficiency while maintaining near-optimal total cost in terms of hiring and transportation costs for caregivers.

In [1], the authors propose a bi-objective model for a daily HHC routing and scheduling problem, aiming to minimize both transportation cost and the number of vehicles used. In their model, the hiring cost for caregivers is not considered since the number of caregivers is predetermined. Each caregiver's route starts and ends at a depot, so the working time is of type 1. Skill level requirements are not considered in their model and the service time for all requests is fixed at 20 minutes. The authors develop two evolutionary algorithm (EA) approaches, both employing a pareto selection scheme.

In [6], the authors define a Markov decision process (MDP) for the HHC problem and they build up a Q-learning model, a model-free reinforcement learning (RL) algorithm, on top of the MDP. The study in [6] employs "mixed time window constraints", which prohibit caregivers to arrive late but allow them to arrive early and wait until the patients' time windows begin. In [6], all caregivers' routes must start and end at the depot, so the working time is of type 1. The proposed Q-learning model effectively balances the trade-off between longer allowable waiting times — offering greater scheduling flexibility — and the inefficiency associated with excessive idle time for caregivers.

In [5], route construction relies heavily on insertion-based scheduling, where patients are sequentially inserted into caregivers' schedules. Since the duration of requests may not perfectly align with the available time slots in caregivers' schedules, gaps may occur between assignments. Thus, caregivers might experience idle time or early arrival for their next assignment. In [5], additional requests are allowed to be submitted during serving, and the HHC system will then on decide, in real-time, whether to accept the requests or not. Reference [5] assumes multiple depots, and for each caregiver, the route must start and end in respective depot. The working time is of type 1. The objective of [5] is to maximize the number of patients served, and it is shown to be able to handle real-time requests submitted by patients during the bi-weekly schedule.

Table 1. Summary of related works for home health care. "(1)" means "(some caregivers may not be hired)", "(2)" means "(all caregivers must be hired)", "Time Window" means "Time Window Constraints", "min." means "minimize", and "max." means "maximize".

References	Set of caregivers	Span	Endpoints	Time Window	Objective
2021 [3]	given in advance (1)	1 week	type 1	hard	min. hiring and transportation costs
2022 [1]	given in advance (2)	1 day	type 1	no	min. vehicles and transportation costs
2023 [6]	given in advance (2)	1 day	type 1	mixed	min. travel and wait times
2024 [5]	given in advance (2)	2 weeks	type 2	mixed	max. number of patients served
this manuscript	adjusted	1 day	type 2	hard	min. hiring and transportation costs

This study is motivated by the following observations.

O1. In [1, 3, 5, 6], the sets of caregivers are given in advance. Thus, there might have redundant caregivers or the number of caregivers might be insufficient.

O2. In [3], the districting approach imposes a rigid structure in which caregivers are restricted to operate solely within their assigned districts, thereby excluding potentially better solutions that span across district boundaries.

O3. None of [3, 5, 6] asks a caregiver to finish the patient's request within the given time window. In other words, [3, 5, 6] may not be patient-friendly since a patient may receive a service outside the expected time window.

O4. In [5, 6], a caregiver is allowed to arrive at the patient's residence "earlier than" the given time window and wait until the time window begins, causing excessive waiting and idle time in the caregiver's schedule. For example, instance C in Figure 5 [6] shows that among 480 minutes of working time, the average waiting time is more than 160 minutes.

In this study, we aim to minimize the cost caused by caregivers' hiring and transportation. Accordingly, the set of caregivers is adjusted (i.e., not given in advanced). The span considered is 1 day, as we believe that it is not difficult to extend our solution to other spans. We consider working time of type 2 since, in our home country, a caregiver's working time begins with the first service and ends with the final service. The time windows are assumed to be hard constraints to ensure patient-friendliness, as defined in Subsection 2.2.

The main contributions of this study are:

- We discard the usage of a predefined caregiver set, which is used in [1, 3, 5, 6]. Instead, we adjust the number of caregivers needed, thus finding a suitable number of caregivers.
- We discard the concept of districting in [3]. Instead, we propose an algorithm, called hiCH, which is based on a hierarchical convex hull approach to minimize both the hiring and the transportation costs of caregivers. To the best of our knowledge, we are the first to adopt convex hull to solve HHC problem.
- We sort the requests based on their levels and time windows, and combine this with hiCH. This results in a new algorithm: D-hiCH.
- We further enhance the solution by combining genetic algorithm (GA) with D-hiCH. By utilizing the solution obtained from D-hiCH as an initial solution, we can accelerate the convergence of GA and search for solution that outperforms D-hiCH.

2 Time Window, Patient-Friendliness, and Problem Formulation

In this section, we first provide the background on time windows. Then, we introduce a new constraint for HHC, aimed at achieving patient-friendliness. Next, we describe our system assumptions. Finally, we formally define the HHC problem addressed in this study.

The following notations are needed. Let $[e_i, l_i]$ denote a time window, where e_i and l_i are the earliest and latest service time requested by patient i, respectively. Additionally, let a_i, b_i, and f_i denote the arrival time, the service beginning time, and the service finishing time of the caregiver assigned to patient i, respectively.

2.1 Time Window Constraint

There are four types of time windows: hard time window (HTW), soft time window (STW), mixed time window (MTW), and hard & soft time window (HTW & STW). In this study, we only consider HTW, in which a caregiver is only allowed to arrive within the time window, i.e., HTW requires that $a_i \in [e_i, l_i]\ \forall i$ and it prohibits both early arrival and tardiness. An illustration of HTW can be found in Figure 2. See [6] for details of other types of time windows.

2.2 Patient-Friendliness Constraint

Let n denote the number of patients. In this manuscript, we assume that all requests are fulfilled under HTW constraints. In order to better respect patient preferences, we introduce the following patient-friendliness constraint:

For each i, $1 \leq i \leq n$, patient i must have $e_i \leq f_i \leq l_i$.

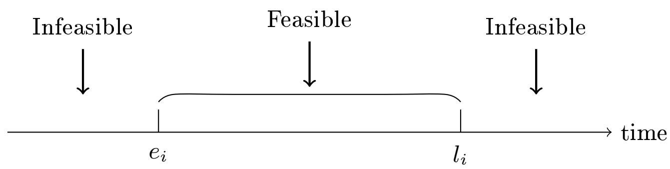


Fig. 2. Hard time window (HTW): arrows indicate caregivers' arrivals.

This constraint enforces that not only must the service begin within the time window, but the completion of service must also fall within it. That is, the entire service must be provided during the patient's available time window.

2.3 System Assumptions

The following system assumptions are used. The span of the considered HHC problem is 1 day. Each patient submits exactly one request. Each patient must be served once and by the same caregiver. HTW is used. Each caregiver can only provide services to patients with required skill levels lower than or equal to the caregiver's skill level. Each caregiver's workload upper bound is $\mathcal{W}$. (In our simulations, $\mathcal{W}$ is 480 minutes.) The service time required by each patient is strictly smaller than $\mathcal{W}$. Each caregiver's working time is of type 2, i.e., the working time begins from starting serving the first patient and ends after finishing serving the last patient. Each caregiver's transportation cost is proportional to the distance traveled.

2.4 Notations

Since the span of the considered HHC problem is 1 day and each patient submits exactly one request, we will not distinguish between patients and their requests. The following notations are introduced for patients. The HHC provider will receive a list of requests submitted by the set of patients $\mathcal{P} = \{1, 2, \ldots, n\}$. The request of patient i, $1 \leq i \leq n$, is represented as a 4-tuple (e_i, l_i, q_i^p, s_i), where $[e_i, l_i]$ is the time window (e_i is the earliest time the service may begin, whereas l_i is the latest time the service must be completed), q_i^p is the qualification level needed to perform the service, and s_i is the service time required. To ensure feasibility, each request (e_i, l_i, q_i^p, s_i) must satisfy $s_i \leq l_i - e_i$. Also, by our assumptions, $s_i < \mathcal{W}$ holds for each patient i, $1 \leq i \leq n$.

The following notations are introduced for caregivers. Let $\mathcal{C}$ denote the set of caregivers. Let $m = |\mathcal{C}|$ be the number of caregivers. Initially, $C = \emptyset$ and $m = 0$. Each caregiver $j \in \mathcal{C}$ is associated with a skill level q_j^c. Notice that a caregiver is allowed to serve for requests with skill levels lower than or equal to her/his qualification level. Let r_j denote the route of visits for caregiver j, let n_j denote the number of patients in r_j (i.e., number of patients assigned to caregiver j), and let r_j^i denote the i-th visited patient on r_j. Let h_j and w_j denote the hiring cost and the working time of caregiver j, respectively.

We model this spatial relationship between all patients' residences by constructing a graph $G = (V, E)$, where V consists of nodes representing residences of patients and E is the set of edges weighted by the distance $d_{u,v}$ between any two patients $u, v \in V$. By convention, G is a complete graph.

2.5 Problem Formulation

The objective of our HHC problem is to minimize the cost incurred by hiring caregivers and transportation between patients' residences for caregivers:

$$\min \quad \underbrace{\sum_{j \in \mathcal{C}} h_j}_{\text{hiring cost}} + \underbrace{\sum_{j \in \mathcal{C}} \sum_{i=1}^{n_j - 1} tr \times d_{r_j^i, r_j^{i+1}}}_{\text{transportation cost}} \tag{1}$$

$$\text{s.t.:} \quad q^p_{r_j^i} \leq q^c_j, \quad \forall\, j \in \mathcal{C},\ \forall\, 1 \leq i \leq n_j \tag{2}$$

$$e_i \leq b_i \leq f_i \leq l_i, \quad \forall\, i \in \mathcal{P} \tag{3}$$

$$\sum_{i=1}^{n_j - 1} d_{r_j^i, r_j^{i+1}} + \sum_{i=1}^{n_j} s_{r_j^i} \ \leq\ \mathcal{W}, \quad \forall\, j \in \mathcal{C} \tag{4}$$

Notice that in our assumption, transportation cost incurred by each caregiver will be proportional to the distance travelled. Thus, the transportation cost incurred from travelling from u to v would be $tr \times d_{u,v}$, where tr is the transportation cost per unit of time travelled. We now explain (1)–(4).

The objective (1) minimizes the sum of hiring cost and transportation cost. Constraint (2) ensures that all patient requests are assigned to caregivers with the appropriate skill levels. Constraint (3) guarantees that all services are provided within the time windows specified by the patients, that is, the patient-friendliness is guaranteed. Finally, constraint (4) ensures that the total working time — including both travelling and service time — for each caregiver does not exceed the working time upper bound.

3 Our Solution to HHC: hiCH

We first give the definition of a convex hull. The *convex hull* of a set of ≥ 3 points is the smallest convex polygon containing all points in the set. Idea of our algorithm hiCH is as follows.

Suppose that j caregivers have already been hired and let $\mathcal{P}$ be the set of yet unvisited patients. Set $j = j + 1$ and let $\mathcal{H}_j$ be the convex hull of points corresponding to the residences of patients in $\mathcal{P}$. Among patients on $\mathcal{H}_j$, we find a patient i_1 with the highest required skill level $q^p_{i_1}$ under

$$i_1 = \operatorname*{argmax}_{i \in \mathcal{H}_j} q^p_i,$$

and hire a caregiver c_j with the corresponding level $q^c_{c_j} = q^p_{i_1}$ to provide the service. In general, if caregiver c_j is serving with current schedule being $r_j = [\, i_1, i_2, \ldots, i_k \,]$, then the $(k+1)$-th patient is selected from the pool $\mathcal{S}^k_j$ below.

Let

$$\mathcal{S}^k_j = \{ i \in \mathcal{P} \mid \underbrace{q^p_i \leq q^c_{c_j}}_{\text{(i)}},\ \underbrace{e_i \leq f_{i_k} + d_{i_k,i}}_{\text{(ii)}},\ \underbrace{f_{i_k} + d_{i_k,i} + s_i \leq l_i}_{\text{(iii)}},\ \text{(iv)} \} \tag{5}$$

where (iv) is

$$\sum_{\ell=1}^{k} s_{i_\ell} + \sum_{\ell=1}^{k-1} d_{i_\ell, i_{\ell+1}} + d_{i_k,i} + s_i \leq \mathcal{W}.$$

Patient i_{k+1} is then selected by

$$i_{k+1} = \operatorname*{argmin}_{i \in \mathcal{S}^k_j} d_{i_k,i}$$

and added to route r_j. The route is eventually completed if we cannot find any new candidate from the pool.

Input to our algorithm hiCH are the set of requests submitted by patients and a matrix containing distances between each pair of patients. Output of our algorithm hiCH are the set of hired caregivers and the routes (i.e., schedules) of caregivers. We now are ready to list our algorithm hiCH in Algorithm 1.

We use the residences of 30 patients whose data are from dataset Ins 1-30-2-s1 in [3] to illustrate how our hiCH successively finds convex hulls and arranges routes for caregivers. See Figure 3 for an illustration. In this figure, the following conventions are adopted.

Nodes colored red are the residences of patients whose caregivers' qualification levels are level 3 (the highest). Nodes colored blue are the residences of patients whose caregivers' qualification levels are level 2 (the median). Nodes colored green are the residences of patients whose caregivers' qualification levels are level 1 (the lowest). Polygons shown by solid lines are the convex hulls. Paths shown by solid lines are the routes of caregivers. In the route for a caregiver, "start" means the "starting residence" and "end" means the "ending residence" of the route. After each route is formed, the visited patients will be marked in grey to indicate that their requests have been fulfilled. This process continues until all patients' requests have been fulfilled.

We summarize the outcome of hiCH for dataset Ins 1-30-2-s1 in [3] in Table 2.

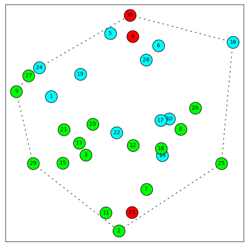
Algorithm 1: hiCH

Input: (1) $\mathcal{P}$: the set of requests submitted by patients,
(2) $(d_{i,j})$: a matrix containing distances between each pair of patients.
Output: (1) $\mathcal{C}$: the set of hired caregivers,
(2) $r_1, r_2, \ldots, r_{|\mathcal{C}|}$: the routes (i.e., schedules) of caregivers.

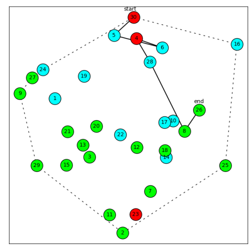
$\mathcal{C} \leftarrow \emptyset$; $j \leftarrow 0$
while $\mathcal{P} \neq \emptyset$ **do**
    $\mathcal{H} \leftarrow$ convex hull of $\mathcal{P}$
    $i \leftarrow \text{argmax}_{k \in \mathcal{H}}\, q^p_k$
    $j \leftarrow j + 1$
    `# hire a new caregiver`
    $\mathcal{C} \leftarrow \mathcal{C} \cup \{c_j\}$, where c_j is a caregiver with skill level $q^c_j = q^p_i$
    $r_j \leftarrow [\,]$ `#` r_j`: route of` c_j
    $w_j \leftarrow 0$ `#` w_j`: workload of` c_j
    `# arrange the route for this new caregiver`
    $r_j \leftarrow r_j + [\,i\,]$ `# add patient` i `to route of` c_j
    $w_j \leftarrow w_j + s_i$ `# provide service to patient` i
    $\mathcal{P} \leftarrow \mathcal{P} \setminus \{i\}$ `# delete patient` i
    poolEmpty $\leftarrow$ False
    while $w_j \leq \mathcal{W}$ **and** NOT poolEmpty **do**
        form pool $\mathcal{S}$ by using (5)
        if $\mathcal{S} \neq \emptyset$ **then**
            $i' \leftarrow \text{argmin}_{i \in \mathcal{S}}\, d_{i,i'}$ `# choose the next patient` i'
            $r_j \leftarrow r_j + [\,i'\,]$ `# add patient` i' `to route of` c_j
            $w_j \leftarrow w_j + d_{i,i'}$ `# travel from patient` i `to patient` i'
            $w_j \leftarrow w_j + s_{i'}$ `# provide service to patient` i'
            $\mathcal{P} \leftarrow \mathcal{P} \setminus \{i'\}$ `# delete patient` i'
            $i \leftarrow i'$
        else
            poolEmpty $\leftarrow$ True
return $\mathcal{C}$ and $r_1, r_2, \ldots, r_{|\mathcal{C}|}$.

Table 2. Outcome of hiCH for dataset Ins 1-30-2-s1 in [3].

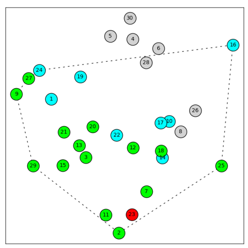
Caregiver j	Skill level	Working time	Service time	Traveling time	Route of the caregiver
$j = 1$	3	453	401	52	$r_1 = [\,30, 5, 6, 4, 28, 8, 26\,]$
$j = 2$	2	433	394	39	$r_2 = [\,16, 10, 14, 17, 12\,]$
$j = 3$	2	463	421	42	$r_3 = [\,19, 27, 9, 1, 13\,]$
$j = 4$	2	455	415	40	$r_4 = [\,24, 21, 20, 3, 22\,]$
$j = 5$	1	473	383	90	$r_5 = [\,29, 15, 11, 2, 25\,]$
$j = 6$	3	299	267	32	$r_6 = [\,23, 7, 18\,]$
Total	13	2576	2281	295	



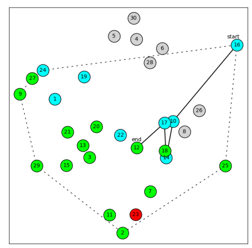
(a) Convex hull $\mathcal{H}_1$.



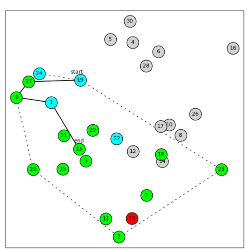
(b) Caregiver 1's route.



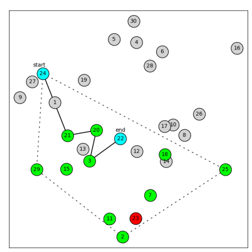
(c) Convex hull $\mathcal{H}_2$.



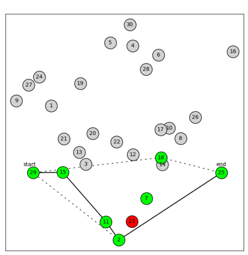
(d) Caregiver 2's route.



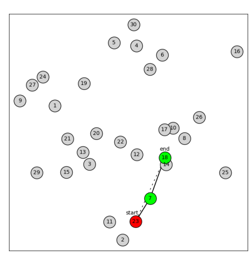
(e) Caregiver 3's route.



(f) Caregiver 4's route.



(g) Caregiver 5's route.



(h) Caregiver 6's route.

Fig. 3. Illustration of the execution of hiCH.

4 Improved Version of hiCH: D-hiCH

We also implement another version of hiCH, called D-hiCH, such that D-hiCH sorts patients' requests in decreasing order of required skill level and after sorting, D-hiCH executes like hiCH. We omit the pseudocode of D-hiCH.

5 GA Version of hiCH: GA + D-hiCH

In [1], the authors demonstrate that evolutionary algorithms (EA) can effectively solve HHC routing and scheduling problems by searching diverse solution spaces while handling complex constraints. Their findings highlight the potential of population-based search methods for HHC, providing the motivation of incorporating GA into our work as an additional refinement tool. Thus, we try to further improve the solution obtained from D-hiCH by integrating a Genetic Algorithm (GA) to explore potentially better assignments and routing structures.

We find that applying GA alone makes it difficult to converge effectively within a reasonable number of generations, and we believe this difficulty arises from the immense size of the search space ($n!$), where n is the number of patients. We also find that if the initial population is fully randomized, as in the conventional GA setup, the algorithm requires a significant amount of time before high-quality individuals appear. To address the aforementioned issues, we replace a portion of the initial population with solutions produced by D-hiCH under minor perturbations. Although our strategy slightly narrows the search space, it injects individuals of relatively high fitness into the population, thereby enabling faster convergence. We refer to this approach as the *search-space-narrowing strategy*. The performance of our search-space-narrowing strategy (or, equivalently, GA + D-hiCH) will be demonstrated in the final table of this manuscript; details of the strategy are described below.

We first encode the HHC problem into chromosomes so that GA can be applied. Given a list of patients, we treat each patient as an index from $\{1, \ldots, n\}$. Thus, each chromosome is represented as a permutation of n elements, which is composed of genes as patients. Assignments and routing are then constructed by scanning the chromosome from left to right: a caregiver is assigned patients sequentially until no more patients can feasibly added to their schedule. The resulting subsequence becomes the caregiver's visiting sequence, and the skill level of the caregiver is the highest required skill level among the patients in the subsequence. The process repeats until all patients have been assigned. The fitness of an individual is the total cost obtained from the entire sequence, namely the sum of caregiver hiring cost and transportation cost for each subsequence.

We find that applying GA alone is difficult to converge effectively. We also find that if the initial population is fully randomized, the algorithm requires significant amount of time before high-quality individuals appear. To address these issues, we thus replace a portion of the initial population with the solution produced from D-hiCH under minor perturbations. In particular, we apply three types of perturbations to the D-hiCH solution:

- Swap — exchange two genes.

- Insert — remove a gene and insert it at a random position.
- Shuffle — randomly permute a selected subsequence.

Each perturbation is applied multiple times, and approximately 25% of the population is generated from each perturbation type. The remaining 25% of the population remains fully random to preserve diversity.

6 Simulation Setup and Results

Our D-hiCH is implemented in Python. Both GA and GA + D-hiCH are implemented in Python using pyGAD. All of our algorithms are executed on a personal computer equipped with an Intel(R) Core(TM) Ultra 7 265KF CPU and 64 GB RAM. We categorize the datasets into two groups based on problem size. For small instances, we use the datasets generated from [3]. For larger instances, we use the real-world datasets provided by the Austrian Red Cross, originally introduced in [2].

- Small instances: $n = 10, 20, 30, 40$.
- Large instances: $n = 75, 100, 125$.

We now give the details of hiring cost, transportation cost, tr, and workload upper bound. First, the hiring cost of a caregiver is based on the caregiver's skill level and we assume: the hiring cost of a caregiver of level 1 is 1000 units, the hiring cost of a caregiver of level 2 is 1.2 times that of a caregiver of level 1, and the hiring cost of a caregiver of level 3 is 1.2 times that of a caregiver of level 2. The transportation cost incurred from travelling from u to v is set to $tr \times d_{u,v}$, where tr is the transportation cost per unit of time travelled. In our simulations, tr is set to 1 unit. The workload upper bound $\mathcal{W}$ for each caregiver is set to 480 units, which includes both the service and the transportation. The value 480 comes from the calculation $8 \times 60 = 480$ if each caregiver works at most 8 hours a day. We now list parameters used in our simulations in Table 3.

Table 3. Parameters and setting.

Parameter	Value
n	a value in $\{10, 20, 30, 40, 75, 100, 125\}$
Hiring cost of a caregiver of level 1	1000 units
Hiring cost of a caregiver of level 2	1200 units
Hiring cost of a caregiver of level 3	1440 units
tr	1 unit
$\mathcal{W}$	480 units

Each group adopts the same GA configurations, described as follows: Parent selection is performed using tournament selection, where a subset of individuals is randomly chosen and the one with the highest fitness is selected as a parent. The size of this subset is determined by the selection number. Selected parents then undergo crossover to produce offspring. We apply order crossover[3], which is an

[3] https://stackoverflow.com/questions/26518393/order-crossover-ox-genetic-algorithm

approach well-suited for permutation-based representations. Mutations may also occur under a predefined mutation probability, using swap mutation, where two randomly chosen genes exchange their positions. Finally, because the population size must remain fixed across generations, only a portion of individuals from the combined pool of parents and offspring are retained. The replacement parameter specifies how many individuals from the current generation survive into the next. Each instance is executed 30 times, and the run containing the individual with the best fitness is selected as the final result. Table 4 summarizes the parameters.

Table 4. Simulation setup for two groups of dataset.

Parameter	$n = 10, 20, 30, 40$	$n = 75, 100, 125$
Population	200	500
Generation	100	300
Selection	tournament	tournament
Selection number	2	5
Crossover	order	order
Mutation	swap	swap
Mutation probability	0.3	0.5
Replacement	50	100

We compare D-hiCH, GA, and GA + D-hiCH. Due to space limitation, we omit the outcome of small instances. Outcome of large instances is shown in Table 5. The followings (i)–(iii) can be observed from Table 5. (i) As n grows larger, GA becomes less effective due to the exponential growth of the search space. (ii) The GA + D-hiCH approach, however, maintains good performance. (iii) Although incorporating high-quality individuals derived from D-hiCH narrows the search space, these individuals help guide the algorithm toward solutions that remain competitive among the three algorithms.

7 Concluding Remarks

In this manuscript, we consider the HHC problem with the goal of minimizing both the total hiring cost and the total transportation cost of caregivers, while maintaining patient-friendliness. We utilize a hierarchical convex hull approach to find a suitable number of caregivers. Based on this approach, we propose three algorithms: hiCH, D-hiCH, and GA + D-hiCH. We conduct extensive simulations on real-world datasets. Simulation results show that GA + D-hiCH is a promising solution for the HHC problem. As a final remark, a preliminary version of this manuscript was in [4].

References

1. M. Belhor, A. El-Amraoui, A. Jemai, and F. Delmotte, "Multiobjective evolutionary algorithm for home health care routing and scheduling problem," in: Proceedings of 2022 8th International Conference on Control, Decision and Information Technologies (CoDIT), Istanbul, Turkey, pp. 1592-1596, 2022.

Table 5. Comparisons of D-hiCH, GA, and GA + D-hiCH with respect to 75, 100, and 125 patients: "n" means "number of patients", each four-tuple (x, y, z, w) represents (hiring cost, transportation cost, total cost, number of caregivers hired), and "†" means "the optimal one".

Dataset in [2]	n	D-hiCH	GA	GA + D-hiCH
U1-n75	75	(17360, 678, 18038, 14)	(15520, 791, 16351, 13)	(15720, 719, 16439, 13)
U2-n75	75	(15920, 716, 16636, 14)	(14680, 834, 15514, 13)	(13480, 642, 14122, 12)
U3-n75	75	(18960, 784, 19744, 16)	(14480, 1043, 15523, 13)	(14720, 759, 15479, 13)
U4-n75	75	(17000, 616, 17616, 14)	(14680, 794, 15474, 13)	(13920, 653, 14573, 12)
U5-n75	75	(19240, 680, 19920, 16)	(15160, 857, 16017, 13)	(14400, 729, 15129, 12)
S1-n75	75	(17800, 1246, 19046, 14)	(17520, 1717, 19237, 15)	(16600, 1158, 17758, 13)
S2-n75	75	(16560, 1342, 17902, 14)	(17080, 1517, 18597, 15)	(15880, 1408, 17288, 14)
S3-n75	75	(19200, 1282, 20482, 16)	(16480, 1561, 18041, 15)	(15680, 1304, 16984, 14)
S4-n75	75	(17520, 1352, 18872, 15)	(18120, 1580, 19700, 16)	(14880, 1332, 16212, 13)
S5-n75	75	(18440, 1320, 19760, 15)	(17560, 1426, 18986, 15)	(15360, 1331, 16691, 13)
Average		(17800, 1001.6†, 18801.6, 14.8)	(16128, 1212, 17344, 14.1)	(15064†, 1003.5, 16067.5†, 12.9†)
U1-n100	100	(25320, 768, 26088, 21)	(21200, 1387, 22587, 18)	(21000, 798, 21798, 18)
U2-n100	100	(22600, 894, 23494, 19)	(21760, 1265, 23025, 19)	(18800, 853, 19653, 16)
U3-n100	100	(19960, 780, 20740, 17)	(20520, 1125, 21645, 18)	(18320, 891, 19211, 16)
U4-n100	100	(21600, 794, 22394, 18)	(20720, 1385, 22105, 18)	(19120, 804, 19924, 17)
U5-n100	100	(25680, 812, 26492, 21)	(23720, 1254, 24974, 21)	(21560, 949, 22509, 19)
S1-n100	100	(28120, 1544, 29664, 23)	(24200, 2258, 26458, 21)	(23640, 1664, 25304, 20)
S2-n100	100	(27000, 1488, 28488, 23)	(23720, 2120, 25840, 21)	(23360, 1336, 24696, 20)
S3-n100	100	(23280, 1660, 24940, 19)	(23960, 2235, 26195, 21)	(20000, 1606, 21606, 17)
S4-n100	100	(23360, 1768, 25128, 20)	(23320, 2258, 25578, 21)	(20880, 1731, 22611, 19)
S5-n100	100	(29880, 1446, 31326, 25)	(26840, 2073, 28913, 23)	(25280, 1579, 26859, 21)
Average		(24680, 1195.4†, 25875.4, 20.6)	(22996, 1736, 24732, 20.1)	(21196†, 1221.1, 22417.1†, 18.3†)
U1-n125	125	(29880, 998, 30878, 25)	(27320, 1830, 29150, 24)	(24560, 1049, 25609, 21)
U2-n125	125	(25400, 996, 26396, 22)	(26800, 1779, 28579, 24)	(23000, 953, 23953, 20)
U3-n125	125	(27760, 1062, 28822, 24)	(26400, 1611, 28011, 23)	(23720, 1050, 24770, 21)
U4-n125	125	(29640, 894, 30534, 25)	(26200, 1739, 27939, 23)	(25160, 952, 26112, 22)
U5-n125	125	(29040, 896, 29936, 24)	(26360, 1727, 28087, 23)	(24520, 1048, 25568, 21)
S1-n125	125	(28640, 1812, 30452, 24)	(30600, 2614, 33214, 27)	(27000, 1756, 28756, 23)
S2-n125	125	(29880, 1924, 31804, 25)	(30880, 2814, 33694, 27)	(27080, 1937, 29017, 23)
S3-n125	125	(30560, 1752, 32312, 25)	(30200, 3081, 33281, 27)	(26320, 1941, 28261, 22)
S4-n125	125	(31200, 1704, 32904, 27)	(30360, 2886, 33246, 27)	(27360, 1866, 29226, 24)
S5-n125	125	(29600, 1814, 31414, 25)	(29800, 2960, 32960, 26)	(26960, 1759, 28719, 23)
Average		(29160, 1385.2†, 30545.2, 24.6)	(28492, 2304.1, 30816.1, 25.1)	(25568†, 1431.1, 26999.1†, 22†)

2. C. Fikar and P. Hirsch, "A matheuristic for routing real-world home service transport systems facilitating walking," Journal of Cleaner Production, vol. 105, pp. 300-310, 2015.
3. E. Nikzad, M. Bashiri, and B. Abbasi, "A matheuristic algorithm for stochastic home health care planning," European Journal of Operational Research, vol. 288, pp. 753-774, 2021.
4. S. Su, *A Novel Algorithm for Assignment and Routing in the Home Health Care Problem*, (Master thesis), National Yang Ming Chiao Tung University, Hsinchu, Taiwan, 2025.
5. Q. Ta-Dinh, T.-S. Pham, M.H. Hà, and L.-M. Rousseau, "A reinforcement learning approach for the online dynamic home health care scheduling problem," Health Care Management Science, vol. 27, pp. 650–664, 2024.
6. T. Zhang, Y. Liu, X. Yang, J. Chen, and J. Huang, "Home health care routing and scheduling in densely populated communities considering complex human behaviours," Computers and Industrial Engineering, vol. 182, 109332, 2023.