

# **Jointly Rostering, Routing, and Reroasting for Home Health Care Services: A Harmony Search Approach with Genetic, Saturation, Inheritance, and Immigrant Schemes**

Chun-Cheng Lin<sup>1</sup>, Lun-Ping Hung<sup>2</sup>, Wan-Yu Liu<sup>3,\*</sup>, and Ming-Chun Tsai<sup>1</sup>

<sup>1</sup>*Department of Industrial Engineering and Management, National Chiao Tung University, Hsinchu 300, Taiwan*

<sup>2</sup>*Department of Information Management, National Taipei University of Nursing and Health Sciences, Taipei 108,  
Taiwan*

<sup>3</sup>*Department of Forestry, National Chung Hsing University, Taichung 402, Taiwan*

## **Abstract**

In home health care (HHC) services, nurses or professional caregivers are dispatched to patients' homes to provide medical care services, such that each patient can stay at home to be treated periodically. The HHC problem consists of the nurse rostering problem (NRP) and the vehicle routing problem with time windows (VRPTW), both of which are NP-hard problems, which are harder or equal to the hardest problem in the NP (nondeterministic polynomial time) problem class and generally cannot be solved efficiently. To the best of our knowledge, the previous algorithmic approaches were designed to separately address NRP and VRPTW of the HHC problem. However, NRP and VRPTW of the HHC problem are intercorrelated, and their respective optimal objectives may be in conflict with each other in many cases. Additionally, the problem generally involves too many constraints to be solved, and most previous works did not address occurrence of sudden incidents in HHC services (e.g., a nurse or a patient suddenly requests for a leave, and a patient suddenly changes the time slot to be treated) such that the original nurse roster could become infeasible. Under constraints of nurse qualifications, working laws, nurse preferences, and vehicle routing, the first model considers nurse rostering and vehicle routing concurrently to minimize

---

\* Corresponding author. Phone: +886-4-22850158. Fax: +886-4-22873628.

*E-mail address:* wyliau@nchu.edu.tw (Wan-Yu Liu)

total costs of nurse overtime and vehicle routing. The second model extends the first model with rerostering caused by occurrence of sudden incidents. Harmony search algorithm (HSA) has been shown to perform better in solving NRPs than conventional metaheuristics, and hence this work proposes an improved HSA with genetic and saturation schemes, in which the solution representation is designed to concurrently determine nurse rostering and vehicle routing. For the second rerostering model, inheritance and immigrant schemes are added to the HSA to adapt to the change caused by occurrence of sudden incidents. Experimental results show that the proposed HSA performs well and can adapt to the change caused by sudden incidents.

*Keywords:* Home health care service, nurse rostering, vehicle routing problem, rerostering, harmony search algorithm, metaheuristic algorithm

## **1. Introduction**

In the home health care (HHC) service, nurses (or professional caregivers) are dispatched to patients' (or caretakers') homes to provide medical treatments or assistance periodically. The home health care problem (HHCP) (Cheng & Rich, 1998) involves the problem of planning nurses' duty rosters (i.e., nurse rostering problem; NRP) (Bagheri et al., 2016) and the problem of allocating nurses to different vehicle routes to meet each patient's time window (i.e., vehicle routing problem with time windows; VRPTW) (Wang et al., 2015). A detailed survey on HHCPs can be found in (Fikar & Hirsch, 2017).

To the best of our knowledge, all the previous works on HHCPs separately addressed VRPTW and NRP of the HHCP. However, VRPTW and NRP of the HHCP are intercorrelated with each other. Additionally, it is crucial to design a nurse roster to decrease nurses' dissatisfaction with fairness and justice of the roster, and further to increase quality of service and decrease medical expenses as well as risks of occupational hazards (Clark et al., 2007; Hallah & Alkhabbaz, 2013). With advance in technologies, lots of medical facilities are equipped with wireless mobile communications and the Internet of things (Marcelli et al., 2007; Bennett-Milburn & Spicer, 2013). With these facilities, hospitals can keep control of patients' conditions. Once a patient has a sudden incident at home, the hospital is able to remotely obtain first-hand information through medical facilities at home, and dispatch nurses to provide immediate services. Such a sudden

patient insertion leads to necessity of in-time rostering.

This work investigates the problem that jointly considers rostering, routing, and rostering for HHC services ( $R^3HHCS$ ), including the following two models. Given information on patients and considering hard and soft constraints in the HHC service, the first model concurrently determines nurse rostering and vehicle routing to patients' homes, with objective of minimizing total costs of nurse overtime and vehicle routing, different from previous works that separately addressed NRP and VRPTW. Given a solution including nurse rostering and vehicle routing for the first model, the second model considers nurse rostering caused by occurrence of sudden incidents in practice. Thus, this rostering model determines a new solution that minimizes not only the objective of the first model but also the difference between the original and the new solutions, because nurses (e.g., those with plans for days-off) would prefer a new solution with not much difference from the original solution.

The HHCP has been shown to be NP-hard (Steeg, 2008), because it includes two NP-hard problems (i.e., NRP and VRPTW), which are harder or equal to the hardest problem in the NP (nondeterministic polynomial time) problem class, and generally cannot be solved efficiently (i.e., in deterministic polynomial time). Hence, it is suitable for applying metaheuristics to solve the HHCP. Hadwan et al. (2013) showed that their proposed harmony search algorithm (HSA) performs better than the genetic algorithm (GA) in solving the NRP. Additionally, Frosolini et al. (2010) proposed a modified harmony search (MHS) algorithm with genetic and saturation schemes for a manufacturing scheduling problem that performs better than the GA and the conventional HSA. For handling dynamics of the concerned problems, this work considers an inheritance scheme to decrease the difference between original and new rosters, and an immigrant scheme (Cheng & Yang, 2010) to increase diversify of the solutions produced. That is, this work proposes an improved HSA with genetic, saturation, inheritance, and immigrant schemes for the  $R^3HHCS$ .

## **2. Related work**

Although the HHCP involves the NRP and the VRPTW, most previous works on the HHCP only focused on addressing the VRPTW in the HHCP. For instance, Cheng and Rich (1998)

considered the VRPTW in which full-time and part-time nurses are considered concurrently to provide the HHC service; and each patient must be served within a fixed time window. They proposed a two-phase construction heuristic for the problem, which first uses a greedy heuristic to establish tours of nurses, and then tightens the nurse rosters.

From the literature, various extensions of VRPTW have received a lot of attention (e.g., Kallehauge et al., 2005; Cordeau et al., 2002; Dohn et al., 2011; Yassen et al., 2015a, 2015b). A line of these extensions is to address the VRPTW in the HHCP. For instance, Thomsen (2006) adopted an algorithm based on insertion heuristic and tabu search (TS) to address the VRPTW with shared visits in the HHC service. Akjiratikarla et al. (2007) adopted a particle swarm optimization algorithm for optimizing the vehicle routing in the HHC service. Bredström and Rönnqvist (2008) adopted a mixed-integer programming model to address the problem of elementary shortest path with time windows in the HHCP. Rasmussen et al. (2012) proposed a branch-and-price approach to address the set partition problem in the HHC service, and regarded the HHCP as a VRPTW. Liu et al. (2013) adopted an algorithm based on GA and TS to address a VRPTW that considers simultaneous delivery and pickup in the HHC service. Fikar and Hirsch (2015) adopted a two-phase algorithm to address the routing and scheduling problem between nurses and patients' homes, in which the first phase identifies potential walking-routes, and the second phase adopts a TS algorithm to search for the optimal routing solution.

However, the HHCP is not just involved with the VRPTW. Thus, some works started considering the NRP in the HHCP. For instance, Bertels and Fahle (2006) integrated linear programming, CP, and metaheuristics to address the NRP in the HHCP, and adopted a two-phase method to address hard and soft time windows as well as nurse availability. Eveborn et al. (2006) adopted a repeated matching algorithm to match different-skill nurses with different-task patients. Some works considered both the NRP and the VRPTW in the HHCP, but separately addressed them. For instance, Nickel et al. (2012) classified the HHCP into nurse rostering, vehicle routing, and nurse rerostering problems, and solved the three problems with different heuristics, respectively. They adopted a two-phase method based on CP and adaptive large neighborhood search for the first nurse rostering problem, a CP metaheuristic for the second vehicle routing problem, and a two-phase method based on insertion heuristic and TS for the nurse rerostering problem caused by new patient insertion. Lessel (2007) classified all visits into clusters according

to geographical locations, nurse skills, and so on. Then, the author proposed a two-phase approach (first dispatching nurses to each cluster, and then solving the VRPTW for each cluster). Allaoua et al. (2013) divided the HHCP into the NRP and the VRPTW, and proposed metaheuristics and mathematical programming methods for them. Maya Duque et al. (2015) adopted a random local search algorithm for the HHCP, and proposed a two-phase approach (first optimizing the service level and then minimizing the total routing distance). Yalçındağ et al. (2016) proposed two-phase decomposition methods with incremental degrees of flexibility guaranteed by joint assignment, scheduling and routing.

Cappanera and Scutellà (2014) considered nurse qualifications to plan nurse rostering and vehicle routing in a given planning horizon. Their work proposed an integer linear programming (ILP) model with objective of balancing workload of nurses and decreasing waiting time between consecutive visits, but did not consider dynamic problems caused by sudden incidents. The differences of their work from this work are as follows: 1) For the problem setting, the objective of their problem is to balance workload of nurses and to decrease waiting time between visits, but the objective of the problem concerned in this work is to minimize the total nurse overtime cost and the total vehicle routing cost. In addition, they did not consider dynamic problems caused by sudden incidents, but this work does. 2) For the methodology, they solved their problem by an ILP model, but this work proposes an MHS metaheuristic algorithm.

Moz and Pato (2004) considered that the hospital periodically announces the nurse roster of next week in advance. For the roster, certain nurse may not be able to fulfill any of the work shifts assigned to this nurse; a new patient could be inserted temporarily but is not served by any nurse (Nickel et al., 2012); certain patient suddenly requests to be absent so that the nurses assigned to this patient are idle. In these situations, the nurse roster should be adjusted. Because the adjusted roster would affect the regular schedules of many nurses, it would be preferred to minimize the original and the adjusted rosters. Moz and Pato (2004) proposed an integer multicommodity flow formulation for the nurse rerostering problem, and they (2007) further proposed a GA for solving the problem more efficiently.

A lot of previous works improved HSA to solve dynamic optimization problems. For instance, Turkey and Abdullah (2014) proposed an improved HSA with random immigrant scheme, memory scheme, and memory based immigrant scheme. From their experimental results, the memory

based immigrant scheme can maintain population diversity, as compared with the other two schemes. Turkey et al. (2014) proposed a multi-population HSA with external archive to efficiently cope with dynamic optimization problems. The differences of their works from this work are as follows: 1) From the problem setting, they evaluated their proposed HSAs on MPB (Moving Peak Benchmark) to cope with feasibility and effectiveness of dynamic problems, but this work investigates novel dynamic problems caused by sudden incidents in the HHCP. 2) To solve dynamic problems, those two works proposed two improved HSAs, which are different from the proposed MHS with genetic, saturation, inheritance, and immigrant schemes in this work.

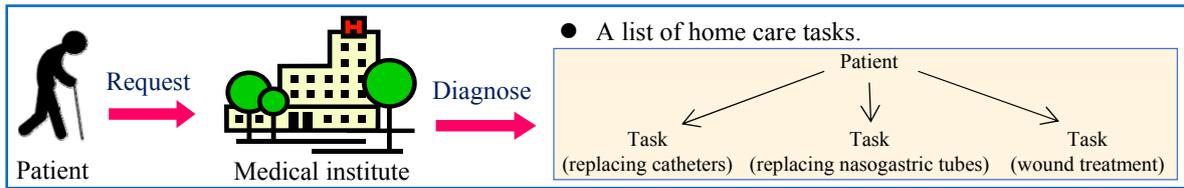
### 3. Nurse Rostering and Routing Problems for Health Care Services

Consider an HHC service system as shown in Fig. 1. Given a number of patients, each of these patients requested an HHC service from the medical institute or service provider (Fig. 1(a)). Next, a doctor in the medical institute diagnosed that this patient had to take a list of HHC tasks (e.g., replacing catheters, replacing nasogastric tubes, wound treatments, and so on) periodically, i.e., weekly in this work. Next, this patient was asked to provide the information to be served, including a fixed time slot of certain day of each week within which all tasks of this patient must be executed (Fig. 1(b)) and the home address (Fig. 1(c)). Note that each day has two time slots: 8:00 AM ~ 2:00 PM, and 2:00 PM ~ 8:00 PM.

This work takes the role of the medical institute that provides the HHC service, and investigates the R<sup>3</sup>HHCS, which includes two problem models, depending on whether a sudden incident occurs. As shown in Fig. 1(d), given the information of patients (including HHC task lists, assigned time slots, and the distance between each home and the medical institute), the first problem model is to determine nurse rosters and vehicle routing of each work shift (time slot) in a week to provide the HHC service to all patients, with the following objectives and constraints:

$$\text{Minimize the total nurse overtime cost} \tag{1}$$

$$\text{and the total vehicle routing cost} \tag{2}$$

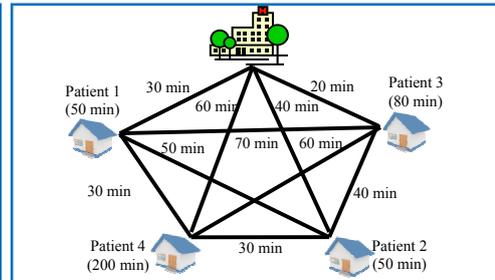


(a) A patient makes a request to the service.

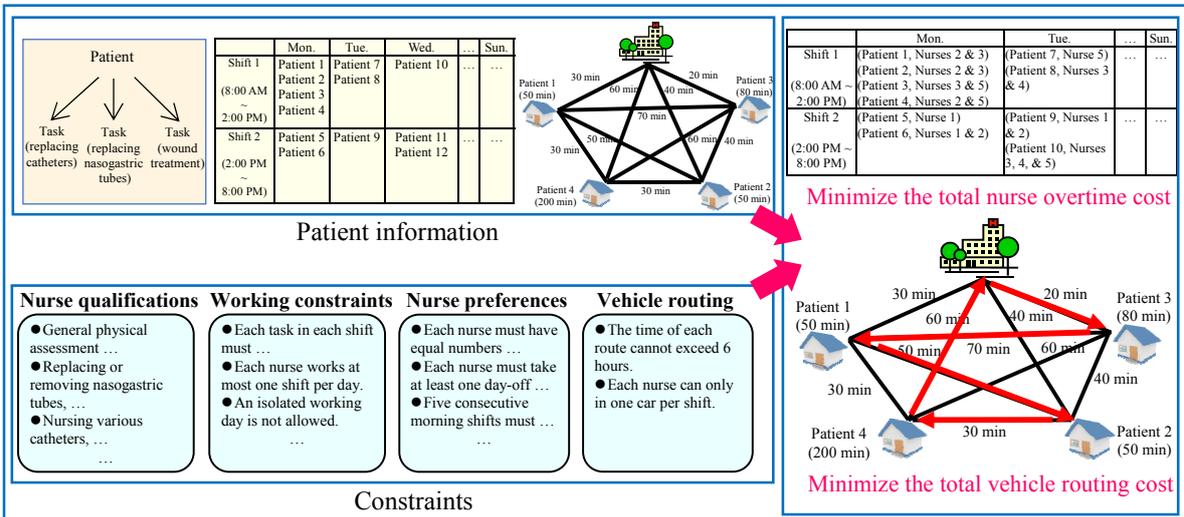
● The patient provides a fixed time slot of certain day of each week to be served.

	Mon.	Tue.	Wed.	...	Sun.
Time slot 1 (8:00 AM ~ 2:00 PM)	Patient 1 Patient 2 Patient 3 Patient 4	Patient 7 Patient 8	Patient 10	...	...
Time slot 2 (2:00 PM ~ 8:00 PM)	Patient 5 Patient 6	Patient 9	Patient 11 Patient 12	...	...

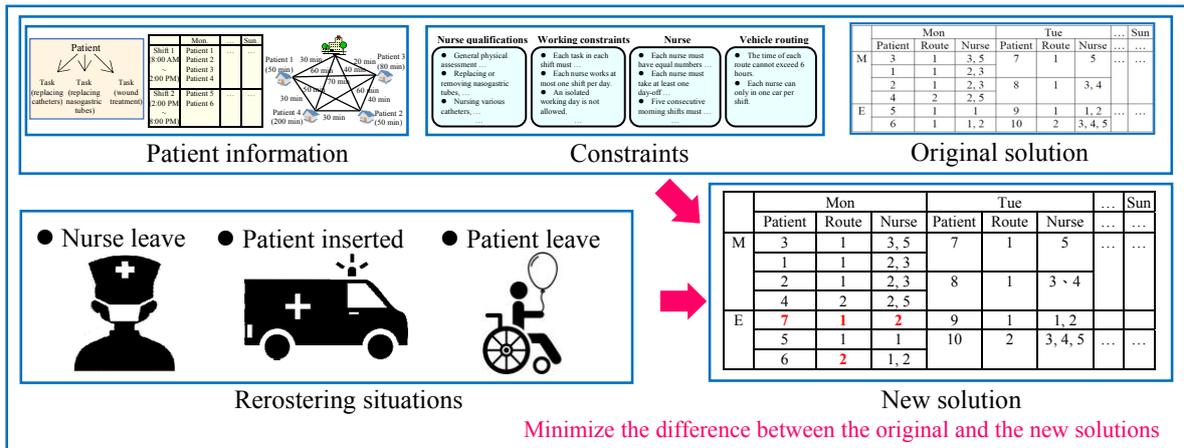
(b) The time slots of all patients.



(c) The patient provides the home address.



(d) The first model (with jointly rostering and routing)



(e) The second model (with jointly rostering, routing, and rerostering)

Fig. 1. Framework of an HHC service system.

s.t.

nurse qualification constraints, (3)

the working constraints defined by laws and contracts, (4)

nurse preference constraints, and (5)

vehicle routing constraints. (6)

The time length of each work shift is 6 hours. Each nurse is supposed to work for at most 6 shifts in one week. Hence, if a nurse works for more than 36 hours in one week, the overtime of this nurse is considered as the extra cost of the medical institute. Therefore, Objective (1) in the above model is to minimize the total overtime costs of all nurses.

Since each nurse cannot work for more than 6 hours, the time length of routing each vehicle that carries nurses is also restricted to 6 hours (Nickel et al., 2012). However, patients may have a high demand for being served in specific time slots (e.g., those in weekdays). Hence, it is necessary to dispatch multiple vehicles in the same time slot. Therefore, Objective (2) is to minimize the total routing costs of multiple vehicles of all time slots.

Since HHC tasks of patients are diversified, not every nurse is omnipotent to execute all categories of HHC tasks. Hence, this work supposes that nurses' professions of executing HHC tasks are divided into five categories (as shown in Table 1), and each category can only be executed by the nurses qualified for this category. Note that each nurse may have more than one profession qualification category listed in Table 1. Therefore, Constraint (3) enforces that each HHC task must be executed by a nurse with the corresponding qualification.

**Table 1.** Five categories of qualifications of nurses.

ID	Statement
1	General physical assessment/examination, nursing wound and pressure sores, and dressing various wound.
2	Replacing or removing nasogastric tubes, replacing inner and outer tracheostomy tubes, and replacing catheters and urine bags.
3	Nursing various catheters, nasogastric tubes, and tracheostomy tubes, and having the nasogastric tube-fed skill.
4	Skill of using bladder lavage, bladder training, as well as large/small-scale enema, and taking/inspecting specimens.
5	Simple rehabilitation, health education, nutrition, and other care guides related to patients

Data source: The HHC service defined by the Department of Health, Taipei City Government.

If each nurse is satisfied with respective work shifts and days-off, and can take sufficient rest, then quality of the HHC service can be promoted, medical expenses of the medical institute can be reduced, and risks of occupational hazards can be lowered (Clark et al., 2007; Hallah & Alkhabbaz, 2013). Therefore, the work shifts of nurses must confirm to labor contracts, regulations of hospital management, and laws of the HHC service (Hadwan et al., 2013). This work regards them as the working constraints in (4), as listed in Table 2. Note that they are considered as hard constraints in this work, because any violation of regulation and laws is a serious problem to the medical institute. Once one of these constraint is violated, the obtained solution is infeasible. Thus, this work penalizes each of these violations with a large cost value (i.e., 1,000).

**Table 2.** The working constraints defined by laws and contracts (Hadwan et al., 2013) (hard constraints).

ID	Constraint statement	Penalty
1	Each task in each shift must be assigned to a nurse.	1,000
2	Each nurse works at most one shift per day.	1,000
3	An isolated working day (i.e., a working day with a day-off before and after the day) is not allowed.	1,000
4	Each nurse cannot work for more than 6 consecutive working days.	1,000

To increase nurses' satisfaction with their work shifts and days-off, Constraint (5) considers nurse preferences, as listed in Table 3, tailored from (Hadwan et al., 2013). Because violation of these constraints just reduces the total nurse satisfaction but does not make the solution infeasible, these constraints are considered as soft constraints in this work. That is, once one of these constraints is violated, this work penalizes each of these violations with a relatively small cost value (i.e., 10).

**Table 3.** The nurse preference constraints (soft constraints).

ID	Constraint statement	Penalty
1	Each nurse must have equal numbers of working days and days off in one week.	10
2	Each nurse must take at least one day-off in the weekends.	10
3	Five consecutive morning shifts must be followed by one day-off	10
4	Five consecutive evening shifts must be followed by one day-off	10

The nurses who are assigned to work for the same shift can be divided into multiple vehicles, but a nurse cannot take two different vehicles in the same work shift. This violation could occur when applying metaheuristics to solve the HHCP (Nickel et al., 2012). Constraint (6) restricts this

violation. Moreover, this work continues the setting in (Nickel et al., 2012) of penalizing this violation as follows. When a nurse takes two different vehicles in the same work shift, an alternative qualified nurse is dispatched so that both the two vehicles can be launched in the same work shift. However, additional alternative nurses lead to additional expenses of the medical institute. Hence, each of these violations is penalized with 10.

As shown in Fig. 1(e), when a sudden incident (e.g., a nurse suddenly requests for a leave, a patient suddenly changes the time slot to be treated, and a patient suddenly requests for a leave) (Moz & Pato, 2004) occurs, the second model of this work is to adjust the original nurse roster (associated with vehicle routing) in time, so that Objectives (1), (2), and the following Objective (7) are minimized.

Minimize the differences between the new nurse roster and the original nurse roster

in which some work shifts have been finished (7)

#### 4. Proposed Algorithms

The HSA (Geem et al., 2001) is a population-based metaheuristic that searches for a solution by simulating an improvisation process (Lee & Geem, 2004) of multiple musicians to produce the best harmony. The MHS algorithm (Frosolini et al., 2010) improves the HSA with a saturation scheme to increase the solution searching efficiency. The HSA has been shown to be successful in addressing the NRP (Hadwan et al., 2013), and the MHS algorithm can further address a manufacturing scheduling problem with multiple objectives (Frosolini et al., 2010). Therefore, this work develops an improved MHS algorithm with genetic, saturation, inheritance, and immigrant schemes to address the R<sup>3</sup>HHCS, which includes scheduling problems.

The MHS algorithm is based on the HSA. In the HSA, a candidate solution (called a *harmony*) for the concerned problem is encoded as a number of decision variable values (called *notes*) for the problem; and a *cost* function corresponding to the objective function of the concerned problem is defined to evaluate performance of a harmony. The HSA iteratively maintains and improves a number *hms* of harmonies, stored in a so-called *harmony memory* (denoted by *HM*) as follows:

$$HM = \begin{bmatrix} x_{1,1} & x_{1,2} & \cdots & x_{1,\eta} & c(x_1) \\ x_{2,1} & x_{2,2} & \cdots & x_{2,\eta} & c(x_2) \\ \vdots & \vdots & \cdots & \vdots & \vdots \\ x_{hms,1} & x_{hms,2} & \cdots & x_{hms,\eta} & c(x_{hms}) \end{bmatrix}$$

where for  $i \in \{1, 2, \dots, hms\}$ ,  $x_i = (x_{i,1}, x_{i,2}, \dots, x_{i,\eta})$  is the  $i$ th harmony consisting of  $\eta$  notes, and  $c(x_i)$  is the cost of harmony  $x_i$ ; for  $j \in \{1, 2, \dots, \eta\}$ ,  $x_{i,j}$  is the  $j$ th note of the  $i$ th harmony. In each iteration of the improvement process of the HSA, a new harmony  $x_{new} = (x_{new,1}, x_{new,2}, \dots, x_{new,\eta})$  is generated as follows:

- 1) Consider each note  $x_{new,i}$  in this new harmony  $x_{new}$ , for  $i \in \{1, 2, \dots, hms\}$ .
- 2) If a random number from  $[0, 1]$  is greater than the harmony memory consideration rate  $HMCR$ , then note  $x_{new,i}$  is set to the value of a random note from  $\{x_{1,i}, x_{2,i}, \dots, x_{hms,i}\}$  (i.e., a random note from the  $i$ th column of the  $HM$  matrix, or say, the  $i$ th node of a random harmony in the  $HM$  matrix); otherwise, note  $x_{new,i}$  is set to a random number within the feasible range of this node.
- 3) If the former condition holds, then the HSA further checks whether another random number from  $[0, 1]$  is greater than the pitch adjustment rate  $PAR$ . If true, let  $x_{new,i} = x_{new,i} + bw \cdot rand(-0.5, 0.5)$  where bandwidth  $bw$  is a small positive number.

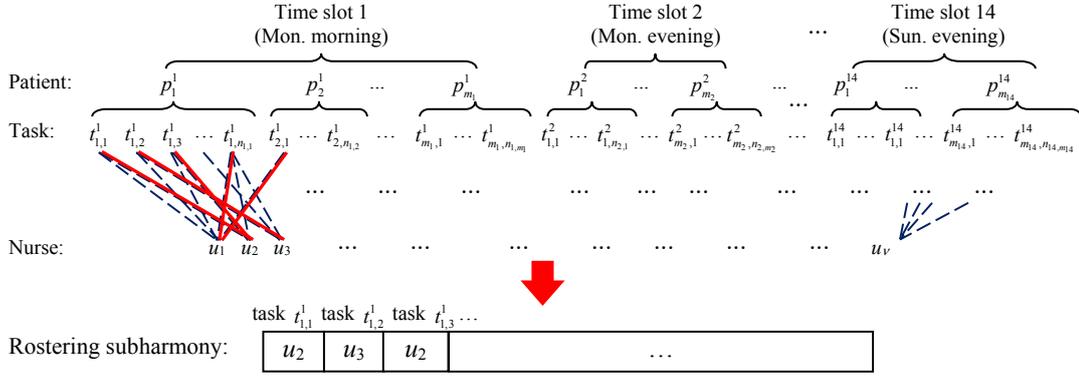
After a new harmony  $x_{new}$  is generated, its cost  $c(x_{new})$  is evaluated. If its cost is less than the cost of the worst harmony  $x_{worst}$  in the  $HM$  matrix, then  $x_{worst}$  is replaced by  $x_{new}$ . The above procedure is repeated until the maximal number of iterations ( $NI$ ) is achieved or all harmonies in the  $HM$  matrix are the same.

This section first introduces the main components of the proposed MHS algorithm for the R<sup>3</sup>HHCS (including solution encoding, solution decoding, cost evaluation, harmony initialization, the inheritance scheme, the immigrant scheme, the saturation scheme, and the operators for generating new harmonies), and finally details the proposed MHS algorithm.

#### 4.1. Solution encoding

Consider an R<sup>3</sup>HHCS with  $m$  patients. Since each day of a week has two time slots (work shifts), there are 14 time slots in one week. For each  $s \in \{1, 2, \dots, 14\}$ , let  $m_s$  denote the number of

patients in the  $s$ th time slot (i.e.,  $m = m_1 + m_2 + \dots + m_{14}$ ), and let  $P_s = \{p_1^s, p_2^s, \dots, p_{m_s}^s\}$  denote the set of the patients who request to be served in the  $s$ th time slot in which  $p_i^s$  denotes the  $i$ th patient in the  $s$ th time slot, as shown in Fig. 2. In the R<sup>3</sup>HHCS, patient  $p_i^s$  is diagnosed to take  $n_{s,i}$  HHC tasks, and  $T_i^s = \{t_{i,1}^s, t_{i,2}^s, \dots, t_{i,n_{s,i}}^s\}$  denotes the set of the tasks for patient  $p_i^s$ , in which  $t_{i,j}^s$  denotes the  $i$ th patient's  $j$ th task in the  $s$ th time slot (see Fig. 2). Let  $n$  denote the number of all tasks (i.e.,  $n = \sum_{s=1}^{14} \sum_{i=1}^{m_s} n_{s,i}$ ). Consider that  $\nu$  nurses (denoted by  $U = \{u_1, u_2, \dots, u_\nu\}$ ) provide the HHC service to the  $m$  patients. For each task, not all nurses are qualified to execute this task. Hence, let  $U_{i,j}^s$  denote the set of the nurses qualified for task  $t_{i,j}^s$ . Note that nurse sets may not be disjoint, i.e., a nurse may have more than one qualification category listed in Table 1.



**Fig. 2.** An example of the bipartite graph for the rostering subharmony.

Since the R<sup>3</sup>HHCS consists of NRP and VRPTW, a candidate solution for the R<sup>3</sup>HHCS is encoded as two parts that respectively address the NRP and the VRPTW. The NRP and the VRPTW are intercorrelated with each other. When planning a vehicle route passing by each patient's home, it is required to check whether a nurse qualified for serving the patient is available to be on duty. Conversely, when planning a roster for nurses serving patients, it is required to check if vehicle routes passing by patients' homes are optimal. Therefore, this work encodes a harmony (candidate solution) for the first problem model of the R<sup>3</sup>HHCS as the following sequence of  $(n + m)$  numbers:

$$\underbrace{\left( \underbrace{\langle \mu_{1,1}^1, \mu_{1,2}^1, \dots, \mu_{1,m_1}^1 \rangle}_{\text{for patient } p_1^1}, \underbrace{\langle \mu_{2,1}^1, \dots, \mu_{2,m_2}^1 \rangle}_{\text{for patient } p_2^1}, \dots, \underbrace{\langle \mu_{m_1,1}^1, \dots, \mu_{m_1,m_1}^1 \rangle}_{\text{for patient } p_{m_1}^1}, \dots, \mu_{m_4,m_4}^{14} \right)}_{\text{Part 1 (rostering subharmony)}} \mid \underbrace{\left( \langle \rho_{1,1}, \rho_{1,2}, \dots, \rho_{1,m_1} \rangle, \langle \rho_{2,1}, \rho_{2,2}, \dots, \rho_{2,m_2} \rangle, \dots, \langle \rho_{14,1}, \rho_{14,2}, \dots, \rho_{14,m_{14}} \rangle \right)}_{\text{Part 2 (routing subharmony)}}.$$

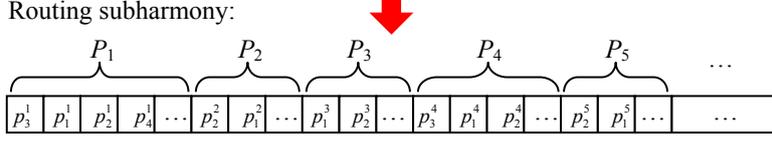
a permutation of  $P_1$       a permutation of  $P_2$       a permutation of  $P_{14}$

The first part is an  $n$ -length vector used for addressing the NRP, in which for  $s \in \{1, 2, \dots, 14\}$ ,  $i \in \{1, 2, \dots, m_s\}$ ,  $j \in \{1, 2, \dots, n_{s,i}\}$ , element  $\mu_{i,j}^s \in U_{i,j}^s$  determines the ID of the nurse that executes task  $t_{i,j}^s$ . The second part is used for addressing the VRPTW, and it consists of 14 permutations  $\langle \rho_{1,1}, \rho_{1,2}, \dots, \rho_{1,m_1} \rangle$ ,  $\langle \rho_{2,1}, \rho_{2,2}, \dots, \rho_{2,m_2} \rangle$ ,  $\dots$ , and  $\langle \rho_{14,1}, \rho_{14,2}, \dots, \rho_{14,m_{14}} \rangle$ , in which  $\langle \rho_{s,1}, \rho_{s,2}, \dots, \rho_{s,m_s} \rangle$  is a permutation of patient set  $P_s$  in the  $s$ th time slot, for each  $s \in \{1, 2, \dots, 14\}$ . For convenience of explanation in the remainder of this paper, the first part is called the *rostering subharmony*, and the second part is called the *routing subharmony*.

On the rostering subharmony, since each task is restricted to be served by a nurse qualified for this task, the relationship between tasks and qualified nurses can be established as a bipartite graph between the set of tasks and the set of nurses. For instance, a bipartite graph between  $n$  tasks and  $\nu$  nurses is shown in Fig. 2, in which if task  $t_{i,j}^s$  can be served by nurse  $u_k$  for any  $s, i, j$ , and  $k$  (i.e., nurse  $u_k$  is qualified to execute task  $t_{i,j}^s$ ), then there is a dotted-line edge between them. That is, each dotted-line edge in this bipartite graph is a feasible assignment for Constraint (3) in the R<sup>3</sup>HHCS. Since each task must be served, a feasible nurse roster in this problem is a subset of edges in this bipartite graph in which each task is covered exactly by an edge. For instance, the red solid-line segments in Fig. 2 represent a feasible nurse roster for Constraint (3). And these segments are corresponding to a rostering subharmony (i.e.,  $(u_2, u_2, u_1, \dots)$  in Fig. 2), which must be feasible for Constraint (3).

On the routing subharmony, since each patient has requested a time slot to be served, it is known that all patients in each time slot of one week must be served, and vehicles must be routed to these patients' homes. Hence, the routing subharmony encodes a routing ordering of the patients in each time slot (i.e., the routing of each time slot is a permutation of the patients that request this time slot). For instance, consider the example in Fig. 3. If the time slot requested by each patient is shown in the upper part of Fig. 3, then a routing subharmony example is given in the lower part of Fig. 3, in which a permutation of the patients for each time slot (i.e.,  $P_s$ ) is listed.

	Mon.	Tue.	Wed.	...
Time slot 1	$p_1^1, p_2^1, p_3^1, p_4^1, \dots$	$p_1^3, p_2^3, \dots$	$p_1^5, p_2^5, \dots$	...
Time slot 2	$p_1^2, p_2^2, \dots$	$p_1^4, p_2^4, p_3^4, \dots$	...	...



**Fig. 3.** An example for the routing subharmony.

When some sudden incident occurs, we enter the second model of the R<sup>3</sup>HHCS. Consider that some work shifts in the original roster have been finished. Let  $\sigma$  denote the index of the first of the remaining time slots. Hence, this work encodes a harmony for the second model as follows:

$$\underbrace{\left( \underbrace{\langle \mu_{1,1}^\sigma, \mu_{1,2}^\sigma, \dots, \mu_{1,n_\sigma}^\sigma \rangle}_{\text{for patient } p_1^\sigma}, \underbrace{\langle \mu_{2,1}^\sigma, \dots, \mu_{2,n_\sigma}^\sigma \rangle}_{\text{for patient } p_2^\sigma}, \dots, \underbrace{\langle \mu_{m_\sigma,1}^\sigma, \dots, \mu_{m_\sigma,n_\sigma}^\sigma \rangle}_{\text{for patient } p_{m_\sigma}^\sigma}, \dots, \mu_{m_4,1}^{14}, \dots, \mu_{m_4,m_4}^{14} \right)}_{\text{Part 1 (rostering subharmony)}} \mid \underbrace{\left( \langle \rho_{\sigma,1}, \rho_{\sigma,2}, \dots, \rho_{\sigma,m_\sigma} \rangle \right)}_{\text{a permutation of } P_\sigma}, \dots, \underbrace{\left( \langle \rho_{14,1}, \rho_{14,2}, \dots, \rho_{14,m_4} \rangle \right)}_{\text{a permutation of } P_{14}} \Big) \cdot$$

#### 4.2. Solution decoding and cost evaluation

When a harmony  $(\mu_{1,1}^1, \dots, \mu_{m_4,1}^{14}, \dots, \mu_{m_4,m_4}^{14} \mid \langle \rho_{1,1}, \dots, \rho_{1,m_1} \rangle, \dots, \langle \rho_{14,1}, \dots, \rho_{14,m_4} \rangle)$  for the first model is generated, the MHS algorithm requires to decode the harmony as a candidate solution for the R<sup>3</sup>HHCS and to calculate a cost value (related to the objective of the R<sup>3</sup>HHCS) to represent the performance of the solution. According to the problem model described in (1)–(6), the cost of the harmony applied in the proposed MHS algorithm for the first model of the R<sup>3</sup>HHCS is denoted by  $c_1$  and is calculated as follows:

$$\begin{aligned}
 & c_1(\mu_{1,1}^1, \dots, \mu_{m_4,1}^{14}, \dots, \mu_{m_4,m_4}^{14} \mid \langle \rho_{1,1}, \dots, \rho_{1,m_1} \rangle, \dots, \langle \rho_{14,1}, \dots, \rho_{14,m_4} \rangle) \\
 &= w_1 \cdot \sum_{k \in \{1, \dots, V\}} \max\{h_k - 36, 0\} / (v \cdot 48) + w_2 \cdot \sum_{s \in \{1, \dots, 14\}} \sum_{r \in \{1, \dots, \gamma_s^s\}} R_r^s / \sum_{s \in \{1, \dots, 14\}} \sum_{i \in \{1, \dots, m_s\}} (2\delta_{0i}^s + \sum_{j \in \{1, \dots, n_{s,j}\}} \tau_{i,j}^s) \\
 &+ 1000 \cdot \varepsilon_1 + 10 \cdot \varepsilon_2 + 10 \cdot \varepsilon_3
 \end{aligned} \tag{8}$$

where  $w_1$  and  $w_2$  are the weights of Objectives (1) and (2), respectively, in which  $0 \leq w_1, w_2 \leq 1$  and  $w_1 + w_2 = 1$ ;  $h_k$  represents the working hours of nurse  $u_k$ ;  $R_r^s$  denotes the routing time of the  $r$ th vehicles in the  $s$ th time slot, and the total number of vehicles  $\gamma_s$  in the  $s$ th time slot is known after

decoding the harmony;  $\delta_{i_1, i_2}^s$  represents the shortest routing time between patients  $p_{i_1}^s$  and  $p_{i_2}^s$ , and for convenience of notation,  $\delta_{0, i}^s = \delta_{i, 0}^s$  represents the shortest routing time between the medical institute and patient  $p_i^s$ ; and  $\tau_{i, j}^s$  represents the time required for task  $t_{i, j}^s$ ;  $\varepsilon_1$  represents the number of violating constraints in Table 2 (i.e., for Constraint (4));  $\varepsilon_2$  represents the number of violating constraints in Table 3 (i.e., for Constraint (5));  $\varepsilon_3$  represents the number of alternative nurses (i.e., for Constraint (6)).

The five terms of the cost function in Equation (8) are explained as follows. Since Objectives (1) and (2) are measured according to different criteria, they are normalized to be within the range [0, 1]. The numerator of the first term (i.e., the total nurse overtime) is to calculate the total working overtime of all nurses, in which each subterm is the working hours  $h_k$  deducted from the nominal working hours of a nurse (i.e., 6 hours  $\times$  6 days = 36 hours), but is zero if the value is negative. The denominator of the first term (i.e., 48 hours) is the number of nurses (i.e.,  $\nu$ ) times the total working hours of one week (i.e., 6 hours  $\times$  2 work shifts  $\times$  7 days = 84 hours) deducted from the nominal working hours of a nurse (i.e., 6 hours  $\times$  6 days). That is, the denominator is the maximal overtime of a nurse in one week. Hence, the first term must fall within the range [0, 1].

The numerator of the second term is to calculate the total routing time of all vehicles in all time slots. The denominator of the second term is the total time of the routing between the medical institute and each patient, plus the treatment time of all tasks of each patient. That is, the denominator is the maximal routing time, when each vehicle routing only passes by a patient's home. Hence, the second term must fall within the range [0, 1]. By weights  $w_1$ ,  $w_2$  and normalization, the sum of the first two terms must fall within the range [0, 1].

On constraints of the concerned problem, since Constraint (3) is always satisfied as designed in the solution encoding in the previous subsection, the cost function in Equation (8) only penalizes each violation of Constraints (4), (5), and (6) (i.e., the latter three terms in Equation (8), respectively). Since Constraint (4) includes only hard constraints, each violation of Constraint (4) is penalized with a large value (i.e.,  $1000 \cdot \varepsilon_1$ ). Since Constraints (5) and (6) are soft constraints, each violation of these constraints is penalized with a small value (i.e.,  $10 \cdot \varepsilon_2$  and  $10 \cdot \varepsilon_3$ ).

Moreover, when some work shifts of a roster have been finished but certain sudden incident

occurs, we enter the second model of the R<sup>3</sup>HHCS. Consider that  $\sigma$  denotes the index of the first of the remaining time slots. To evaluate a harmony  $(\mu_{1,1}^\sigma, \dots, \mu_{m_{14}, n_{14}, m_{14}}^{14} | \langle \rho_{\sigma,1}, \dots, \rho_{\sigma, m_\sigma} \rangle, \dots, \langle \rho_{14,1}, \dots, \rho_{14, m_{14}} \rangle)$  at the second model, the cost  $c_2$  of the harmony is calculated as follows:

$$\begin{aligned}
& c_1(\mu_{1,1}^\sigma, \dots | \langle \rho_{\sigma,1}, \dots, \rho_{\sigma, m_\sigma} \rangle, \dots, \langle \rho_{14,1}, \dots, \rho_{14, m_{14}} \rangle) \\
&= w_1 \cdot \sum_{k \in \{1, \dots, \nu\}} \max\{h_k - 36, 0\} / (\nu \cdot 48) + w_2 \cdot \sum_{s \in \{\sigma, \dots, 14\}} \sum_{r \in \{1, \dots, \gamma_s\}} R_r^s / \sum_{s \in \{\sigma, \dots, 14\}} \sum_{i \in \{1, \dots, m_s\}} (2\delta_{0i}^s + \sum_{j \in \{1, \dots, n_{s,j}\}} \tau_{i,j}^s) \\
&+ 1000 \cdot \varepsilon_1 + 10 \cdot \varepsilon_2 + 10 \cdot \varepsilon_3 + 10 \cdot \varepsilon_4
\end{aligned} \tag{9}$$

where  $\varepsilon_4$  is the number of differences between the original and new rosters. Note that Equation (9) is similar to Equation (8), in which the major difference between them is to introduce  $\varepsilon_4$ , used for cost of penalizing the difference between the original roster and the new roster in which some work shifts have been finished.

The algorithm of decoding a harmony and evaluating its cost is given in Algorithm 1, which is suitable for both the two problem models of the R<sup>3</sup>HHCS, and is explained as follows. First of all, Lines 1–3 read the first part of the input rostering subharmony to determine and record the duty roster of each nurse. Next, Line 4 – 9 calculate the working hours  $h_k$  of each nurse  $u_k$  in all remaining work shifts of one week. Next, Line 10 initializes the three variables  $\varepsilon_1$ ,  $\varepsilon_2$ , and  $\varepsilon_4$  used for recording the penalty costs for Constraints (4), (5), and (6), respectively, to zero. Next, Lines 11 – 21 check whether each constraint listed in Table 2 and 3 (i.e., Constraints (4) and (5), respectively) is violated, and increase  $\varepsilon_1$  or  $\varepsilon_2$  by one if a corresponding constraint is violated. Specifically, Lines 13, 14, and 16 are designed for Statements 2, 3, and 4 in Table 2, respectively. Note that Statement 1 in Table 2 always holds because the solution encoding has been designed for this statement. Lines 17 – 20 are designed for Statements 1 – 4 in Table 3, respectively.

---

**Algorithm 1** Cost Evaluation()

---

**Input:** Harmony  $x = (\mu_{1,1}^\sigma, \dots | \langle \rho_{\sigma,1}, \dots, \rho_{\sigma, m_\sigma} \rangle, \dots, \langle \rho_{14,1}, \dots, \rho_{14, m_{14}} \rangle)$ , in which if the problem is at the first model, then  $\sigma = 1$  and  $c = c_1$ ; otherwise,  $\sigma$  depends on the time when the sudden incident occurs and  $c = c_2$ .

**Output:** Cost  $c(x)$ .

- 1: **for** each  $(s, i, j)$  where  $s \in \{\sigma, \sigma + 1, \dots, 14\}$ ,  $i \in \{1, \dots, m_s\}$ ,  $j \in \{1, \dots, n_{s,j}\}$  **do**
- 2: Record that the nurse corresponding to note  $\mu_{i,j}^s$  of the input rostering subharmony is on duty in the  $s$ -th time slot
- 3: **next for**
- 4: **for**  $k = 1$  to  $\nu$  **do**
- 5:  $h_k = 0$
- 6: **for**  $s = \sigma$  to 14 **do**
- 7: If nurse  $u_k$  is on duty in the  $s$ -th time slot, then  $u_k = u_k + 6$
- 8: **end for**

```

9: next for
10:  $\varepsilon_1 = \varepsilon_2 = \varepsilon_3 = 0$ 
11: for  $k = 1$  to  $\nu$  do
12:   for  $s = 1$  to  $7$  do
13:     If nurse  $u_k$  works for two shifts in the  $s$ th day of one week, then  $\varepsilon_1 = \varepsilon_1 + 1$ 
14:     If nurse  $u_k$  works in the  $s$ th day but is off in the former and following days, then  $\varepsilon_1 = \varepsilon_1 + 1$ 
15:   next for
16:   If nurse  $u_k$  works for at least one shift in any of 6 or 7 days in one week, then  $\varepsilon_1 = \varepsilon_1 + 1$ 
17:   If the number of working days of nurse  $u_k$  is not equal to that of nurse  $u_1$ , then  $\varepsilon_2 = \varepsilon_2 + 1$ 
18:   If nurse  $u_k$  is not off in both the 6th and 7th days, then  $\varepsilon_2 = \varepsilon_2 + 1$ 
19:   If nurse  $u_k$  has worked for 5 consecutive morning shifts but is not off after, then  $\varepsilon_2 = \varepsilon_2 + 1$ 
20:   If nurse  $u_k$  has worked for 5 consecutive evening shifts but is not off after, then  $\varepsilon_2 = \varepsilon_2 + 1$ 
21: next for
22: for  $s = \sigma$  to  $14$  do
23:   Set  $i = 1$ ,  $r = 1$  and  $R_r^s = 0$ 
24:   while  $i \leq m_s$  do
25:     Let  $p_{i'}^s$  denote the patient corresponding to note  $\rho_{s,i}$  of the input routing subharmony
26:      $i = i + 1$ 
27:     if  $R_r^s = 0$  then
28:        $R_r^s = 2\delta_{0i'}^s + \sum_{j \in \{1, \dots, n_{s,i'}\}} \tau_{i',j}^s$ 
29:     else
30:       Let  $p_\ell^s$  denote the last patient visited in the routing of the  $r$ th vehicle
31:        $\Gamma = R_r^s - \delta_{0,\ell}^s + \delta_{\ell,i'}^s + \delta_{i',0}^s + \sum_{j \in \{1, \dots, n_{s,i'}\}} \tau_{i',j}^s$ 
32:       if  $\Gamma \leq 6$  hours then
33:          $R_r^s = \Gamma$ 
34:       else
35:          $r = r + 1$ 
36:          $R_r^s = 0$ 
37:          $i = i - 1$ 
38:       end if
39:     end if
40:   end while
41:   Set  $\gamma_s = r$ 
42:   for  $k = 1$  to  $\nu$  do
43:      $\varphi =$  the times nurse  $u_k$  appears in all the  $\gamma_s$  vehicle routings
44:     If  $\varphi \geq 2$ , then  $\varepsilon_3 = \varepsilon_3 + (\varphi - 1)$ 
45:   end if
46: next for
47: Calculate  $c(x)$  according to Equation (8) or (9) and output it

```

---

Next, Lines 22 – 46 decode the routing subharmony and calculate the total vehicle routing time as well as the penalty cost for Constraint (6). Line 22 considers each time slot  $s$  of the remaining time slots. Line 23 initializes the index  $i$  of the  $m_s$  patients in the  $s$ th time slot to one, the vehicle number  $r$  to one, and the routing time of the  $r$ th vehicle in the  $s$ th time slot (i.e.,  $R_r^s$ ) to zero. Next,

the while loop in Lines 24 – 40 iterates index  $i$  unless it is greater than the upper bound  $m_s$ . In each iteration, Line 25 decodes the patient corresponding to note  $\rho_{s,i}$  of the input routing subharmony and denotes it as  $p_i^s$ . Although Line 26 increases the iteration number  $i$  by one, the remainder of this loop may decrease it by one (Line 35) if this patient is not considered to be included in the current vehicle routing. Line 27 checks whether  $R_r^s = 0$  (i.e., whether the routing time of the  $r$ th vehicle is not calculated yet). If true, the  $r$ th vehicle considers only the routing from the medical institute to patient  $p_i^s$ 's home and then directly back to the medical institute. Hence, the routing time  $R_r^s$  is the twice of the routing time between the medical institute and patient  $p_i^s$ 's home (i.e.,  $2\delta_{0i'}^s$ ) plus the time used for executing all tasks in patient  $p_i^s$ 's home (i.e.,  $\sum_{j \in \{1, \dots, n_{s,i}\}} \tau_{i',j}^s$ ).

Otherwise, Lines 30 and 31 calculate the routing time  $\Gamma$  when patient  $p_i^s$ 's home is added as the last station of the routing of the  $r$ th vehicle. Line 32 checks if the routing  $\Gamma$  does not exceed the time period of a work shift (i.e., 6 hours). If true, Line 33 sets the current routing time to  $\Gamma$ . Otherwise, it is required to launch a new empty vehicle (i.e., to increase the vehicle number by one in Line 35), and its routing time is set to zero (i.e.,  $R_r^s = 0$  in Line 36). Since patient  $p_i^s$ 's home is not considered in the last routing, the iteration number goes back in Line 37 so that this patient can be considered in the next iteration.

After the while loop, Line 41 records the final vehicle number  $r$  as  $\gamma_s$ . The for loop in Lines 42 – 45 considers whether each nurse  $u_k$  violates Constraint (6). Line 43 counts the times nurse  $u_k$  appears in all the  $\gamma_s$  vehicle routings. In Line 44, if the times are at least two, the times more than one is considered as a violation of Constraint (6). Finally, with all information, Line 47 calculates cost  $c_1(x)$  according to Equation (8) or cost  $c_2(x)$  according to Equation (9) and outputs it.

#### 4.3. Initializing harmonies for the first model

On initializing a harmony, the proposed algorithm randomly generates a rostering subharmony according to the bipartite graph corresponding to the concerned problem (Fig. 2), and randomly generates the permutations for 14 time slots in the routing subharmony (Fig. 3). Hence, this initial harmony must be feasible for Constraint (3).

#### 4.4. Initializing harmonies for the second model

When certain sudden incident occurs, the R<sup>3</sup>HHCS enters the second model, i.e., the roster should be adjusted to respond to this sudden incident. This work applies the inheritance scheme and the random immigrant scheme to initialize harmonies for the second model.

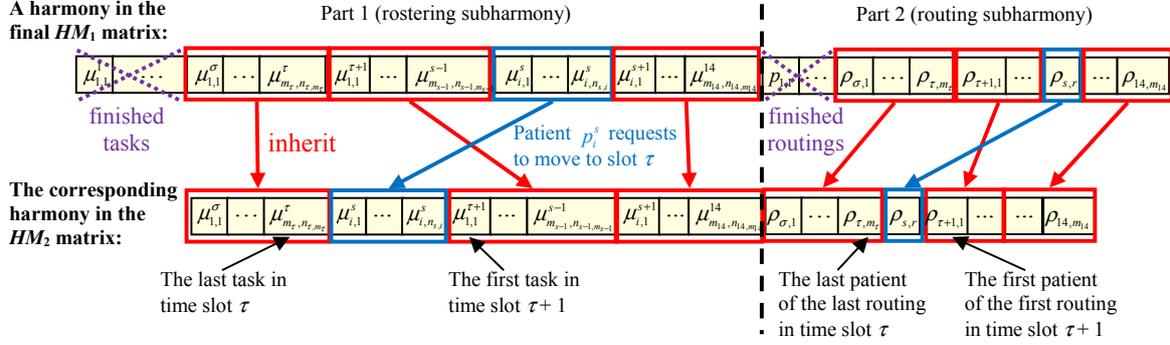
#### 4.4.1. The inheritance scheme

Let  $HM_1$  and  $HM_2$  denote the  $HM$  matrices for the first and second models, respectively. The inheritance scheme is to initialize all harmonies in the  $HM_2$  matrix by inheriting those in the  $HM_1$  matrix. Consider three sudden incidents: 1) a nurse suddenly requests for a leave (e.g., the nurse suddenly got a sick or had a private matter, so that the nurse cannot finish some shifts in the original roster), 2) a patient suddenly changes the service time slot to be treated (e.g., the patient has a sudden emergency and needs an HHC service in time), and 3) a patient suddenly requests for a leave (e.g., the nurse suddenly has a private matter, so that the patient will not be at home during the time slot arranged in the original roster). The inheritance scheme for the three incidents are introduced respectively as follows.

In Incident 1), consider the absence of certain nurse  $u_k$  in time slot  $s$ . Then, it leads to removal of the corresponding dotted-line edge between patients and nurse  $u_k$  in the partite graph for the problem (Fig. 2). The inheritance scheme copies all notes of each harmony in the  $HM_1$  matrix to  $HM_2$  matrix except for the notes for finished tasks and routings. Additionally, for the rostering subharmony, if certain task was served by the absent nurse  $u_k$  in time slot  $s$ , then the task is changed to be served by another qualified nurse according the bipartite graph.

In Incident 2), consider that the  $i$ th patient  $p_i^s$  in time slot  $s$  requests to change to be served in time slot  $\tau$ . Without loss of generality, consider that  $\tau < s$ , in the sense that the patient would like to be served earlier. The inheritance scheme generates a harmony in the  $MH_2$  matrix by copying and swapping some parts of each harmony in the  $HM_1$  matrix, as illustrated in Fig. 4, which is explained as follows. For the rostering subharmony, the notes for finished tasks are removed; the notes for the tasks for patient  $p_i^s$  are inserted between the last task in the required time slot  $\tau$  and the first task in time slot  $\tau + 1$ ; and positions of the other notes remain the same. For the routing subharmony, the notes for finished routings are removed; the note for patients  $p_i^s$  (denoted by  $\rho_{s,r}$ , i.e., the  $r$ th patient of the permutation of set  $P_s$ ) is inserted between the last patient of the last

routing in time slot  $\tau$  and the first patient of the first routing in time slot  $\tau + 1$ ; and positions of the other notes remain the same.



**Fig. 4.** Illustration of the inheritance scheme when the  $i$ th patient  $p_i^s$  in time slot  $s$  requests to change to be served in time slot  $\tau$ .

In Incident 3), consider that the  $i$ th patient  $p_i^s$  in time slot  $s$  requests to be absent in this week or quit the HHC service. The inheritance scheme for Incident 3) is similar to that for Incident 2) except the notes for patients  $p_i^s$ 's tasks and routing are removed so that they are not needed to be copied to the  $HM_2$  matrix.

In light of the above, since the initial  $HM$  matrix for the second model inherits the final  $HM$  matrix for the first model, the difference between the rosters for the two models would be less, so that the requirement of the second model could be met.

#### 4.4.2. The immigrant scheme

By the inheritance scheme, all harmonies in the initial  $HM$  matrix for the second model are inherited from the final  $HM$  matrix for the first model. However, almost all harmonies in the final  $HM$  matrix for the first model would have similar appearance, so that the result of each run of the algorithm for the second model may not have much difference. Hence, to increase diversify of the solutions produced by the proposed algorithm, a part of the initial harmonies are replaced by random harmonies. In general, these harmonies are called *random immigrants* (Cheng & Yang, 2010), because they are generated without any knowledge of the concerned problem.

#### 4.5. Generating new harmonies

Each iteration of the proposed algorithm generates a new harmony and uses it to replace the worst harmony in the  $HM$  matrix if it has a lower cost. This subsection introduces the following four operators of generating new harmonies used in the proposed algorithm.

- 1) *Selecting historical harmonies*: The operators of respectively generating new rostering and routing subharmonies are different. In the rostering subharmony, each note is independent with each other. Hence, the note  $\mu_{i,j}^s$  for patient  $p_i^s$ 's  $j$ th task in the  $s$ th time slot in the new rostering subharmony is set to the value of a random note in the same column of the  $HM$  matrix. In the routing subharmony, notes are not independent. Hence, the operator for the routing subharmony is to randomly select a routing harmony in the  $HM$  matrix, and to directly use the whole routing harmony as the new routing subharmony.
- 2) *Generating random harmonies*: For the rostering subharmony, each note is to randomly select a nurse corresponding to a feasible edge in the bipartite graph in Fig. 2. For the routing subharmony, the routing subharmony for the first model consists of 14 permutations, whereas that for the second model consists of  $(15 - \sigma)$  permutations, where  $\sigma$  denotes the first time slot of the second model. Hence, for the first model, 14 permutations of sets  $P_1, P_2, \dots, P_{14}$ , respectively, are generated randomly; and for the second model,  $(15 - \sigma)$  permutations of sets  $P_\sigma, P_{\sigma+1}, \dots, P_{14}$ , respectively, are generated randomly.
- 3) *Roulette wheel selection and two-point crossover operator*: In the GA (Holland, 1992), selection and crossover operators play an important role in searching for solutions. Various selection and crossover operators existed. After many experimental trials of different operator combinations, the roulette wheel selection (Holland, 1992) and the two-point crossover operator (Goldberg & Lingle, 1985) perform best for the concerned problem. Note that we have implemented one-point, two-point, and uniform crossover operators. After a lot of experimental trials, the two-point crossover operator performs best for the concerned problem. It is speculated that the proposed harmony encoding consists of two subharmonies, and two-point cuts for each of the two subharmony performs well. The roulette wheel selection is a well-known operator, and we do not change any setting of this operator. Hence, only the two-point crossover operator is detailed as follows. Without loss of generality, only the case for the first model of the  $R^3HHCS$  is explained, whereas that for the second model is

similar. First, two harmonies  $x_{old1}$  and  $x_{old2}$  are selected using two times of roulette wheel selection. Next, two cut points (i.e., two positions of the harmony) are decided randomly. Let  $\lambda_1$  and  $\lambda_2$  denote the two cut-point positions;  $\lambda_1, \lambda_2 \in \{1, 2, \dots, n+m\}$ ; and  $\lambda_1 < \lambda_2$ . According to positions  $\lambda_1, \lambda_2$  in the harmony, there are three cases as follows:

- If  $\lambda_1 < \lambda_2 \leq n$  (i.e., both are in the rostering subharmony), consider representing harmony  $x_{old1}$  as  $x_{old1}(1, \lambda_1) \mid x_{old1}(\lambda_1 + 1, \lambda_2) \mid x_{old1}(\lambda_2 + 1, n+m)$  where  $x_{old1}(a, b)$  represents the  $a$ th to the  $b$ th notes of harmony  $x_{old1}$ ; and similarly, harmony  $x_{old2}$  as  $x_{old2}(1, \lambda_1) \mid x_{old2}(\lambda_1 + 1, \lambda_2) \mid x_{old2}(\lambda_2 + 1, n+m)$ . Since both cut points are in the rostering subharmony (in which each note is independent with each other), the two offspring harmonies after the crossover operator are  $x_{new1} = x_{old1}(1, \lambda_1) \mid x_{old2}(\lambda_1 + 1, \lambda_2) \mid x_{old1}(\lambda_2 + 1, n+m)$  and  $x_{new2} = x_{old2}(1, \lambda_1) \mid x_{old1}(\lambda_1 + 1, \lambda_2) \mid x_{old2}(\lambda_2 + 1, n+m)$ , i.e., the second part of each harmony is swapped with each other.
- If  $\lambda_1 < n < \lambda_2$  (i.e.,  $\lambda_1$  is in the rostering subharmony and  $\lambda_2$  is in the routing subharmony), consider that  $\lambda_2$  is in the permutation of the patients in the  $\tau$ th time slot. Then, harmony  $x_{old1}$  is represented as  $x_{old1}(1, \lambda_1) \mid x_{old1}(\lambda_1 + 1, \lambda_2) \mid x_{old1}(\lambda_2 + 1, \sum_{s=1}^{\tau} m_s) \mid x_{old1}(\sum_{s=1}^{\tau} m_s + 1, n+m)$ , and harmony  $x_{old2}$  is represented similarly. Then, after the crossover operator,  $x_{new1} = x_{old1}(1, \lambda_1) \mid x_{old2}(\lambda_1 + 1, \lambda_2) \mid (x_{old1}(\sum_{s=1}^{\tau-1} m_s, \sum_{s=1}^{\tau} m_s) \setminus x_{old2}(\sum_{s=1}^{\tau-1} m_s, \sum_{s=1}^{\tau} m_s)) \mid x_{old1}(\sum_{s=1}^{\tau} m_s + 1, n+m)$ , in which the third part represents  $x_{old1}(\sum_{s=1}^{\tau-1} m_s, \sum_{s=1}^{\tau} m_s)$  excluding  $x_{old2}(\sum_{s=1}^{\tau-1} m_s, \sum_{s=1}^{\tau} m_s)$ , because the permutation of a patient set does not allow any repeated patient note. Symmetrically,  $x_{new2} = x_{old2}(1, \lambda_1) \mid x_{old1}(\lambda_1 + 1, \lambda_2) \mid (x_{old2}(\sum_{s=1}^{\tau-1} m_s, \sum_{s=1}^{\tau} m_s) \setminus x_{old1}(\sum_{s=1}^{\tau-1} m_s, \sum_{s=1}^{\tau} m_s)) \mid x_{old2}(\sum_{s=1}^{\tau} m_s + 1, n+m)$ .
- If  $n \leq \lambda_1 < \lambda_2$  (i.e., both are in the routing subharmony), consider that  $\lambda_1$  and  $\lambda_2$  are in the permutations of the patients in the  $\tau_1$ th and  $\tau_2$ th time slots, respectively. Without loss of generality, assume that  $\tau_1 \neq \tau_2$ ; otherwise, the operator can be done similarly. Harmony  $x_{old1}$  is represented as  $x_{old1}(1, \lambda_1) \mid x_{old1}(\lambda_1 + 1, \sum_{s=1}^{\tau_1} m_s) \mid x_{old1}(\sum_{s=1}^{\tau_1} m_s + 1, \sum_{s=1}^{\tau_2-1} m_s) \mid x_{old1}(\sum_{s=1}^{\tau_2-1} m_s + 1, \lambda_2) \mid x_{old1}(\lambda_2, n+m)$ , and harmony  $x_{old2}$  is represented similarly. Then, after the crossover operator,  $x_{new1} = x_{old1}(1, \lambda_1) \mid (x_{old1}(\sum_{s=1}^{\tau_1-1} m_s, \sum_{s=1}^{\tau_1} m_s) \setminus x_{old2}(\sum_{s=1}^{\tau_1-1} m_s, \sum_{s=1}^{\tau_1} m_s)) \mid$

$$x_{\text{old2}}(\sum_{s=1}^{r_1} m_s + 1, \sum_{s=1}^{r_2-1} m_s) \mid (x_{\text{old1}}(\sum_{s=1}^{r_2-1} m_s, \sum_{s=1}^{r_2} m_s) \setminus x_{\text{old2}}(\sum_{s=1}^{r_2-1} m_s, \sum_{s=1}^{r_2} m_s)) \mid x_{\text{old1}}(\lambda_2, n + m) .$$

Symmetrically,  $x_{\text{new2}} = x_{\text{old2}}(1, \lambda_1) \mid (x_{\text{old2}}(\sum_{s=1}^{r_1-1} m_s, \sum_{s=1}^{r_1} m_s) \setminus x_{\text{old1}}(\sum_{s=1}^{r_1-1} m_s, \sum_{s=1}^{r_1} m_s)) \mid$

$$x_{\text{old1}}(\sum_{s=1}^{r_1} m_s + 1, \sum_{s=1}^{r_2-1} m_s) \mid (x_{\text{old2}}(\sum_{s=1}^{r_2-1} m_s, \sum_{s=1}^{r_2} m_s) \setminus x_{\text{old1}}(\sum_{s=1}^{r_2-1} m_s, \sum_{s=1}^{r_2} m_s)) \mid x_{\text{old2}}(\lambda_2, n + m) .$$

- 4) *Pitch adjusting operator*: In the HSA, the pitch adjusting operator is a local search operator. It first selects a note randomly, and adjusts it slightly according to its position in the harmony. If the note is in the rostering subharmony, then the note is a random feasible nurse for the corresponding task in the bipartite graph for the problem (Fig. 2). If the note is in the routing subharmony, the note is swapped with a random note in the permutation in which the note is.

Note that except for selection and crossover operators are from the GA, the other operators of generating new harmonies above are from the conventional HSA.

#### 4.6. The saturation scheme

When the HSA runs for some iterations, it may have too many similar harmonies in some cases so that it may converge early. The MHS algorithm (Frosolini et al., 2010) additionally considers a saturation scheme to avoid the solution searching ability from rigidity. This scheme introduces a saturation variable denoted by  $Sat$ , which is updated at each iteration of the main loop according to the following formula:

$$Sat = \frac{\text{number of harmonies with a copy in the } HM \text{ matrix}}{hms - 1} . \quad (10)$$

Obviously, the  $Sat$  value falls within the range  $[0, 1]$ . In addition, this saturation scheme considers two thresholds  $Sat_H$  and  $Sat_L$  so that  $0 < Sat_L < Sat_H < 1$ . According to the  $Sat$  value and two thresholds  $Sat_H$  and  $Sat_L$ , two variables  $LPRR$  (low portion recovery rate) and  $PAR$  (pitch adjusting rate) are adjusted. Possible values of  $LPRR$  includes 0,  $LPRR_L$ , and  $LPRR_H$ ; and  $0 < LPRR_L < LPRR_H < 1$ . The possible values of  $PAR$  includes  $PAR_L$ , and  $PAR_H$ ; and  $0 < PAR_L < PAR_H < 1$ .

The MHS algorithm bases on the  $Sat$  value to decide to execute either the conventional HSA operators (i.e., selecting historical harmonies and generating random harmonies detailed in the last subsection) or the GA operators (i.e., roulette wheel selection and two-point crossover operators detailed in the last subsection). In general, the former HSA operators tend to generate more diversified harmonies than the latter GA operators. More specifically, this decision of using HSA

or GA operators to some degree of diversity is made through the  $LPRR$  and  $PAR$  values, determined by the  $Sat$  value. A smaller  $LPRR$  value leads to a higher probability of executing the HSA operators, rather than the GA operators. Furthermore, a larger  $PAR$  value makes the HSA operators generate more diversified harmonies. The MHS algorithm has the following three cases according to the value of  $Sat$  to decide the  $LPRR$  and  $PAR$  values:

- 1) If  $Sat > Sat_H$  (meaning that more harmonies have copies in the  $HM$  matrix), then the algorithm sets  $LPRR$  to 0 and  $PAR$  to  $PAR_H$ , so that the latter iterations of the algorithm only execute the HSA operators in a more diversified way.
- 2) If  $Sat \leq Sat_L$  (meaning that less harmonies have copies in the  $HM$  matrix), then the algorithm sets  $LPRR$  to  $LPRR_L$  and  $PAR$  to  $PAR_L$  so that there is a higher probability for executing the HSA operators but in a less diversified way, whereas there is a lower probability for executing the GA operators.
- 3) If  $Sat_L < Sat \leq Sat_H$  (meaning that the case is between the above two cases), then the algorithm sets  $LPRR$  to  $LPRR_H$  and  $PAR$  to  $PAR_L$  so that there is a higher probability for executing the GA operators, whereas there is a lower probability for executing the HSA operators in a less diversified way.

#### 4.7. The proposed MHS algorithm

With the designs described in the above subsections, the MHS algorithm is given in Algorithm 2, which is explained in detail as follows. The MHS algorithm is used for both models of the  $R^3HHCS$ . Hence, the difference of the algorithms between the two models is that if the problem is at the first model, then  $\sigma = 1$ ; otherwise,  $\sigma$  depends on the time when the sudden incident occurs.

---

#### **Algorithm 2** Proposed MHS()

---

**Required:** Let  $\sigma$  denote the index of the first time slot to be considered. If the problem is at the first model, then  $\sigma = 1$  and  $c = c_1$ ; otherwise,  $\sigma$  depends on the time when the sudden incident occurs and  $c = c_2$ .

- 1: Set parameters  $Sat_L$ ,  $Sat_H$ ,  $HMCR$ ,  $LPRR_L$ ,  $LPRR_H$ ,  $PAR_L$ , and  $PAR_H$
- 2: **if** the problem is at the first model **then**
- 3:   Initialize  $hms$  random feasible harmonies
- 4: **else**
- 5:   Initialize  $P_r \cdot hms$  random immigrant harmonies and  $(1 - P_r) \cdot hms$  harmonies using the inheritance scheme
- 6: **end if**
- 7: Evaluate the  $hms$  harmonies generated above and store them to the  $HM$  matrix
- 8: Calculate the saturation variable  $Sat$  according to Equation (10)
- 9: **for**  $i = 1$  to the maximal number of iterations ( $NI$ ) **do**

```

10: if  $Sat \leq Sat_L$  then
11:    $LPRR = LPRR_L$  and  $PAR = PAR_L$ 
12: else if  $Sat_L < Sat \leq Sat_H$  then
13:    $LPRR = LPRR_H$  and  $PAR = PAR_L$ 
14: else if  $Sat > Sat_H$  then
15:    $LPRR = 0$  and  $PAR = PAR_H$ 
16: end if
17:  $\mathcal{X} = \emptyset$ 
18: if  $rand(0, 1) > LPRR$  then
19:   for each  $(s, i, j)$  where  $s \in \{\sigma, \sigma + 1, \dots, 14\}$ ,  $i \in \{1, \dots, m_s\}$ ,  $j \in \{1, \dots, n_{s,i}\}$  do
20:     if  $rand(0, 1) < HMCR$  then
21:        $\mu_{i,j}^s$  is generated by the harmony memory consideration for rostering subharmony
22:     else
23:        $\mu_{i,j}^s$  is assigned to a random nurse that is qualified for task  $t_{i,j}^s$ 
24:     end if
25:   next for
26:   if  $rand(0, 1) < HMCR$  then
27:      $\langle \rho_{\sigma,1}, \rho_{\sigma+1,1}, \dots, \rho_{14,m_4} \rangle$  is generated by the harmony memory consideration for routing subharmony
28:   else
29:      $\langle \rho_{\sigma,1}, \rho_{\sigma+1,1}, \dots, \rho_{14,m_4} \rangle$  is generated as random permutations of all the patients in the  $(15 - \sigma)$  time slots
30:   end if
31:   Let  $x_{new} = (\mu_{1,1}^\sigma, \dots, \langle \rho_{\sigma,1}, \dots, \rho_{14,m_4} \rangle)$ 
32:    $\mathcal{X} = \mathcal{X} \cup \{x_{new}\}$ 
33: else
34:   for  $i = 0$  to  $\lfloor hms/2 \rfloor$  do
35:     Select two harmonies from the  $HM$  by the roulette wheel selection operator
36:     Perform the two-point crossover operator on the two selected harmonies to generate two harmonies  $x_{new1}$ 
37:     and  $x_{new2}$ 
38:      $\mathcal{X} = \mathcal{X} \cup \{x_{new1}, x_{new2}\}$ 
39:   next for
40: end if
41: for each harmony  $x_{new}$  in  $\mathcal{X}$  do
42:   If  $rand(0, 1) < PAR$ , then perform the pitch adjustment operator on each note of harmony  $x_{new}$ 
43:   Evaluate  $c(x_{new})$ 
44:   Let  $x_{worst}$  be one of  $\{x_1, x_2, \dots, x_{hms}\}$  so that  $c(x_{worst}) = \max\{c(x_1), c(x_2), \dots, c(x_{hms})\}$ 
45:   If  $c(x_{new}) < c(x_{worst})$ , then replace  $x_{worst}$  by  $x_{new}$ 
46: next for
47: Update the  $Sat$  value according to Equation (10)
48: next for
49: Decode the best harmony in the  $HM$  matrix as the final solution
50:

```

---

Algorithm 2 has the following eight main steps:

Step 1. Initialize all parameters (Line 1).

- Step 2. Initialize the  $HM$  matrix according to the problem model type (1 or 2) (Lines 2 – 7).
- Step 3. Update the  $Sat$  value (Lines 8, 48).
- Step 4. Set  $LPRR$  and  $PAR$  values according to the  $Sat$  value (Lines 10 – 16).
- Step 5. If a random number between 0 and 1 is less than  $HMCR$ , then conduct the HMCR operator from the HSA (Lines 19 – 34); otherwise, conduct selection and crossover operators from the GA (Lines 36 – 40).
- Step 6. Conduct the  $PAR$  operator from the HSA (Line 43).
- Step 7. Update the  $HM$  matrix (Lines 44 – 46).
- Step 8. Repeat Steps 3 to 7 until reaching the maximal number of iterations.

Details of Algorithm 2 are as follows. Line 1 initializes all parameters. Next, Lines 2 – 7 initialize  $hms$  harmonies, evaluate them (Subsection 3.2), and store them in the  $HM$  matrix. If the problem is at the first model, then these harmonies are initialized randomly (Subsection 3.3) because we have no information on solutions at the beginning. Otherwise, when considering the second model, since the final  $HM$  matrix for the first model stores  $hms$  harmonies, a part of the  $hms$  harmonies are generated by inheriting from the final  $HM$  matrix for the first model (Subsubsection 3.4.1). Note that the  $hms$  harmonies for the first model are suitable for the first model. If all harmonies at the second model are also from those for the first model, they cannot reflect the sudden incident at the second model. Hence, the other harmonies are generated randomly (Subsubsection 3.4.2).

Line 8 calculates the saturation variable  $Sat$  according to Equation (10). Next, Lines 9 – 49 are the main loop of the algorithm, which runs until the maximal iteration number is achieved. Lines 10 – 16 set the  $LPRR$  and  $PAR$  values according to the  $Sat$  value in the three cases as described in Subsection 3.6. Line 17 sets a set  $\chi$  that stores new harmonies generated in this iteration, and initializes it to an empty set. Line 18 checks if a random number from  $[0, 1]$  is greater than the  $LPRR$  value. If true, Lines 19 – 34 executes the HSA operations (i.e., operators 1 and 2 in Subsection 3.5); otherwise, Lines 36 – 40 execute the GA operators (i.e., operator 3 in Subsection 3.5).

The former HSA operators generate a new harmony  $x_{\text{new}}$ , in which Lines 19 – 26 generate the rostering subharmony, and Lines 27 – 32 generate the routing subharmony. Since each note in the rostering subharmony is independent with each other, each note is generated independently by either the harmony memory consideration operator (see operator 1 in Subsection 3.5) in Line 21 or a random qualified nurse (see operator 2 in Subsection 3.5) in Line 23, depending on whether a random number from  $[0, 1]$  is smaller than the  $HMCR$  value. Since notes in the routing subharmony are permutations of patients, the entire routing subharmony must be generated in a whole. Hence, Line 27 checks if a random number from  $[0, 1]$  is smaller than the  $HMCR$  value. If true, Line 28 generates the routing subharmony by the harmony memory consideration operator (see operator 1 in Subsection 3.5); otherwise, Line 31 generates it as a random feasible one (see operator 2 in Subsection 3.5). Line 33 constitutes the two subharmony generated above as a new harmony  $x_{\text{new}}$ . Next, Line 34 includes harmony  $x_{\text{new}}$  to set  $\chi$ .

Lines 36 – 40 generate two harmonies using the roulette wheel selection operator (Line 37) and the two-point crossover operator (Line 38) from the GA (see operator 3 in Subsection 3.5), and include the generated two new harmonies to set  $H$  (Line 39).

Line 42 considers each new harmony  $x_{\text{new}}$  in set  $\chi$ . Line 43 checks if a random number from  $[0, 1]$  is smaller than the  $PAR$  value. If true, then it conducts the pitch adjusting operator (see operator 4 in Subsection 3.5) on harmony  $x_{\text{new}}$ . Line 44 evaluates cost  $c(x_{\text{new}})$ . Line 45 finds the worse harmony  $x_{\text{worst}}$  from the  $HM$  matrix. If  $x_{\text{new}}$  is better than  $x_{\text{worst}}$ , then Line 46 replaces  $x_{\text{new}}$  by  $x_{\text{worst}}$ .

## 5. Experimental Design and Analysis

With the design detailed in the previous section, this work implements a prototype of the MHS algorithm in C++ programming language. To evaluate performance of this prototype, this section first gives the experimental design of two problem instances and the parameter settings of the algorithm. Then, experimental and statistical results of the proposed MHS algorithm for the first model of the R<sup>3</sup>HHCS are analyzed. Finally, the performance of the proposed MHS algorithm for the second model of the R<sup>3</sup>HHCS is verified in four scenarios.

### 5.1. Experimental design

The experiments of this work are conducted on two different-scale problem instances with parameter settings in Table 4, in which some parameters are referred to (Nickel et al., 2012); whereas the others are determined after a lot of experimental trials or are referred to the practice. The two problem instances are available online<sup>†</sup>. The first problem instance considers 53 patients living in a rural area that request 287 HHC tasks. Since the location is a rural area (i.e., with a larger geographical area), both the caring time of patients and the routing time among patients are longer. The second problem instance considers 95 patients living in a city that request 361 HHC tasks. Both the caring time of patients and the routing time among patients are shorter than the first problem instance.

**Table 4.** Parameter settings of two problem instances.

Instance	Number of patients	Number of tasks	Number of tasks per patient	Task time (minutes)	Number of nurses	Number of qualifications per nurse	Location range (km <sup>2</sup> )
Rural	53	287	3 ~ 7	10 ~ 60	21	2 ~ 5	10 × 10
City	95	361	2 ~ 5	10 ~ 60	38	2 ~ 5	5 × 5

In the first rural problem instance, 287 tasks are randomly distributed to 53 patients under the constraint that each nurse has 3 to 7 tasks. In the second city problem instance, 361 tasks are randomly distributed to 95 patients with the same constraint. The duration of each task is set to a random number from the range between 10 and 60 minutes. According to Table 1, each nurse is randomly assigned with 2 to 5 qualifications. For evaluating the routing time among patients' homes, all patients' homes are randomly distributed within a range of 10 km × 10 km in the rural problem instance (resp., 5 km × 5 km in the city problem instance), where the medical institute is located in the center position, from analogy of the VRP (Ioria & Riera-Ledesma, 2015). Assuming that the average vehicle speed is 40 km/hr (Soysala et al., 2015), all routing distances can be transformed into routing time.

The difficulty of finding feasible solutions depends on the number of nurses. However, a large number of nurses cause a high operating cost. Hence, we conducted the experiments of various parameter combinations to find the most appropriate number of nurses for the two problem instances. When setting the *HM* size (*hms*) to 60 and the number of iterations to 100,000 (Zammori

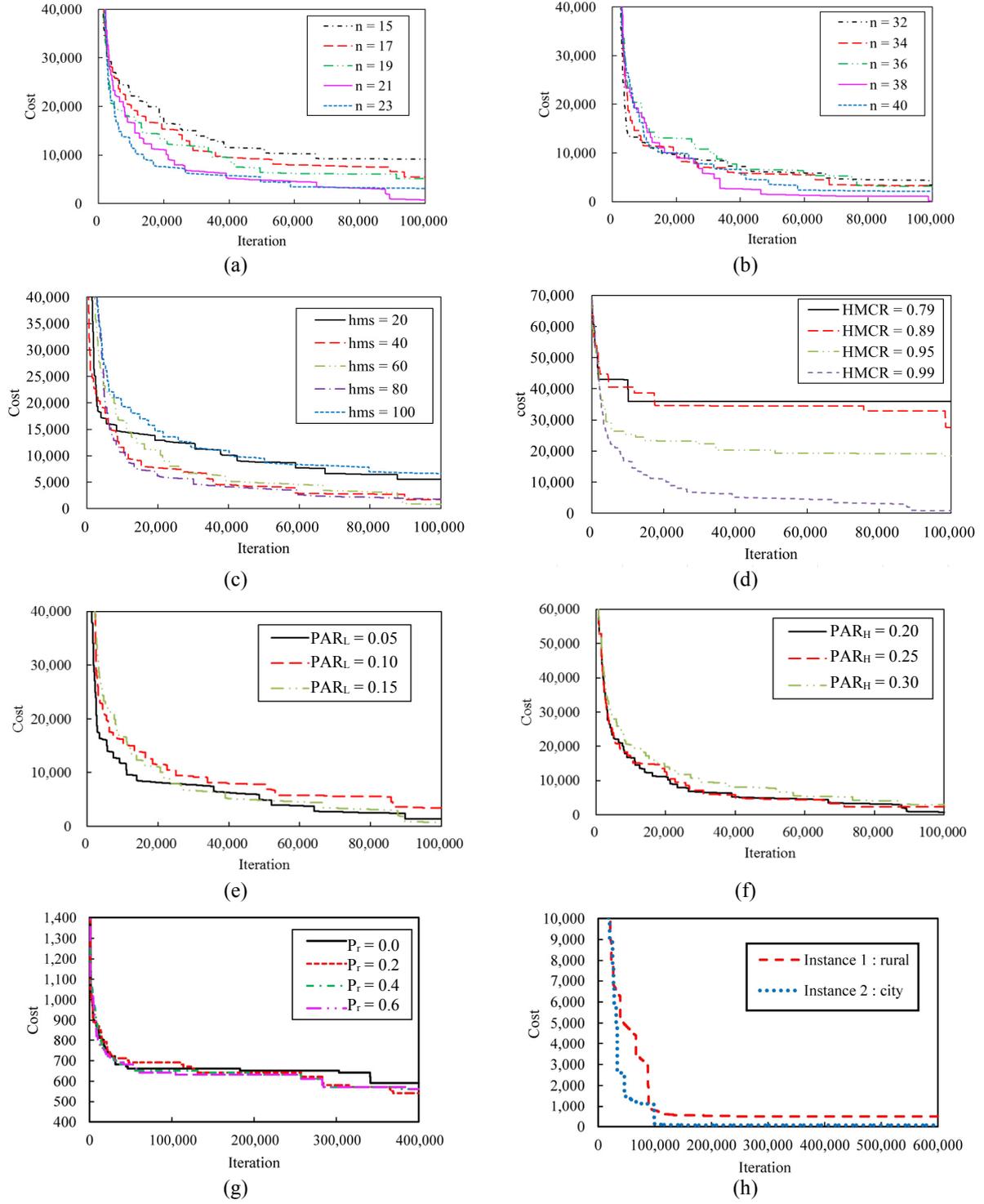
<sup>†</sup> <http://web.it.nctu.edu.tw/~cclin321/problemSet/R3HHCP-instance.rar>

et al., 2014), the experimental results with different numbers of nurses for the two problem instances are shown in Figs. 5(a) and 5(b), respectively, in which the vertical axis represents cost  $c_1$  and the horizontal axis represents the number of iterations. From the costs at the earlier iterations in Fig. 5(a), when the number of nurses ( $n$ ) is larger, the cost is lower because the number of constraint violations is less. However, from the costs at the latter iterations in Fig. 5(a), the result when  $n = 21$  is better than that when  $n = 23$ , because more nurses could increase the probability of violating the 3rd hard constraint in Table 2 (i.e., the probability of causing isolated working days becomes higher because number of nurses is more). Hence, the number of nurses for the first problem instance is set to 21 in latter experiments. By similar discussion on Fig. 5(b), the number of nurses for the second problem instance is set to 38.

The parameters of the proposed MHS algorithm are determined according to their sensitivity analysis. In what follows, only the parameters  $hms$ ,  $HMCR$ ,  $PAR_L$ ,  $PAR_H$ ,  $P_r$ , and  $NI$  are discussed, because the other parameters have less influence on the algorithm and hence are set according to (Zammori et al., 2014). In addition, the sensitivity analyses of these parameters on the two problem instances are similar, and hence only the analysis for the first problem instance is provided.

The experimental results using the proposed HMS algorithm under different  $hms$  values (in which other parameters are set according to (Zammori et al., 2014)) are shown in Fig. 5(c). From this figure, when  $hms$  increases (i.e., more diversified harmonies are considered), the cost tends to decrease in general. However, the results when  $hms$  is greater than 60 do not become better. It is speculated that the information amount more than a threshold may not help improve the solution. Hence, parameter  $hms$  is set to 60 in later experiments. The previous work in (Lee & Geem, 2004) suggested that the  $HMCR$  value should be in the range  $[0.79, 0.99]$ ; the values of  $PAR_L$  and  $PAR_H$  should be within the range  $[0.1, 0.3]$ . From the experimental comparison on these values within their ranges, we set  $HMCR = 0.99$ ,  $PAR_L = 0.15$ , and  $PAR_H = 0.2$  in later experiments. By similar discussion on Fig. 5(g) for the ratio of immigrant harmonies in the  $HM$  matrix ( $P_r$ ), we set  $P_r = 0.2$ .

The plots of costs versus number of iterations ( $NI$ ) when using the proposed MHS algorithm for the two problem instances are given in Fig. 5(h). When  $NI$  is no greater than 100,000, the costs for both problem instances have much improvement. However, the cost has less improvement when  $NI$  is between 100,000 and 400,000, and has almost no improvement after 400,000 iterations. Hence, we set  $NI = 400,000$ .



**Fig. 5.** Experimental results under different (a) numbers of nurses in the first problem instance, (b) numbers of nurses in the second problem instances, (c)  $hms$  values, (d)  $HMCR$  values, (e)  $PAR_L$  values, (f)  $PAR_H$  values, (g)  $P_r$  values, and (h)  $NI$  values.

With the above discussions, the parameter settings used in the proposed algorithm are listed in Table 5.

**Table 5.** The parameters used in the proposed MHS algorithm.

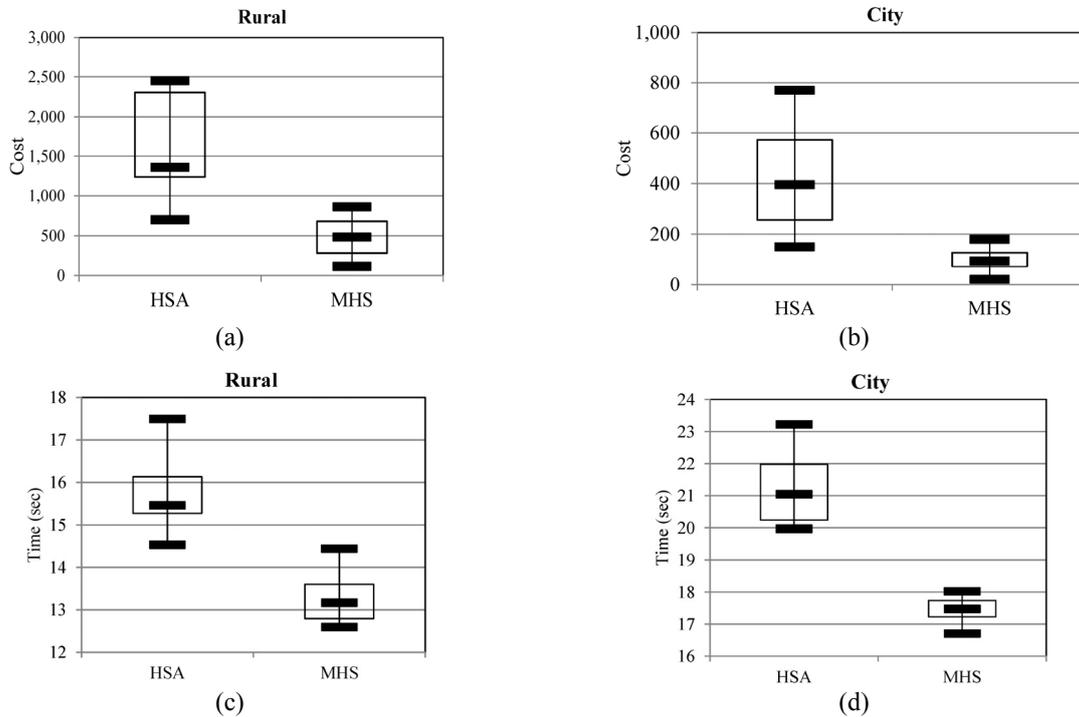
Parameter	Value
Number of iterations ( $NI$ )	400,000
Harmony memory size ( $hms$ )	60
Harmony memory considering rate ( $HMCR$ )	0.99
Low pitch adjusting rate ( $PAR_L$ )	0.15
High pitch adjusting rate ( $PAR_H$ )	0.20
Low large portion recovery rate ( $LPRR_L$ )	0.09
High large portion recovery rate ( $LPRR_H$ )	0.90
Low saturation ( $Sat_L$ )	0.20
High saturation ( $Sat_H$ )	0.80
Ratio of random immigrants ( $P_r$ )	0.20

### 5.2. Experimental analysis for the first model

To fairly analyze the quality of the experimental results, this work executes 20 runs of the HSA and the proposed MHS algorithm for the two problem instances. The boxplots of costs and computational time of the experimental results of the two algorithms are given in Fig. 6, in which Figs. 6(a) and 6(b) compare the costs between two problem instances, whereas Figs. 6(c) and 6(d) compare the computational times between two problem instances.

On the cost results, for the first rural problem instance, the mean of the cost results of 20 runs of the HSA is 1,365.498 (from the left bar of Fig. 6(a)), and 1 hard constraint and about 36 soft constraints are violated on average (i.e., the results using the HSA are infeasible). The mean of the cost results of 20 runs of the proposed MHS algorithm is 485.497 (from the right bar of Fig. 6(a)), and about 48 soft constraints are violated on average. Although the latter violates more soft constraints, it does not violate any hard constraint, and hence the results are feasible. For the second city problem instance, the HSA has a mean of 395.497 and violates about 39 soft constraints on average (from the left bar of Fig. 6(b)), whereas the proposed MHS algorithm has a mean of 95.496 and violates about 9 soft constraints on average (from the right bar of Fig. 6(b)). From these results, the MHS algorithm performs much better than the HSA on average. Furthermore, we observe that the costs in Fig. 6(a) are generally higher than those in Fig. 6(b). It is speculated that the former case (i.e., the rural problem instance) has a smaller number of nurses

than the latter case (i.e., the city problem instance) and hence tends to violate more soft constraints.



**Fig. 6.** Boxplots of costs and computational time of the experimental results between the HSA and the proposed MHS algorithm for two problem instances. Note that the scale of each boxplot is different.

On the computational time results, for the first rural problem instance, the mean of the cost results of 20 runs of the HSA is 15.463 s (from the left bar of Fig. 6(c)), and that of the proposed MHS algorithm is 13.167 s (from the right bar of Fig. 6(c)). For the second city problem instance, the mean of the HSA is 21.045 s (from the left bar of Fig. 6(b)), whereas that of the proposed MHS algorithm is 17.472 s (from the right bar of Fig. 6(b)). From these results, the proposed MHS algorithm is executed more efficiently than the HSA. After observing the result of each iteration in detail, this work discovers that the GA operators (which consider harmonies in a whole) in the proposed MHS algorithm are executed more efficiently than the HSA operators (which consider each note of each harmony). Since the MHS algorithm includes both of GA and HSA operators, it is executed more efficiently than the HSA (with only HSA operators).

To analyze whether the results of using the HSA and the proposed MHS algorithm have significant difference, this work conducts Wilcoxon signed-rank tests on costs and computational

times of the experimental results using the HSA and the proposed MHS algorithm, with a 99% confidence level, as shown in Table 6. All the P values of the two measures between two algorithms for the two problem instances are less than 0.01. That is, the results between the two algorithms for the two problem instances have significant difference.

**Table 6.** The P values from Wilcoxon signed-rank tests for the costs and computational times between the HSA and the proposed MHS algorithm for the two problem instances.

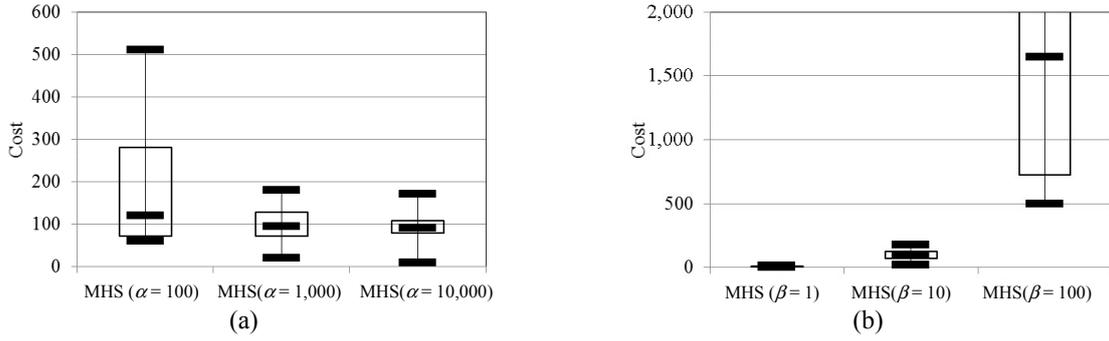
Instance	Cost	Computational time
Rural	$1.145 \times 10^{-8}$	$6.293 \times 10^{-8}$
City	$1.233 \times 10^{-7}$	$6.302 \times 10^{-8}$

In light of the above, the proposed MHS performs remarkably better than the HSA in terms of both cost and computational time.

To analyze the effects of the costs of penalizing hard and soft constraints, Equation (9) is rewritten as follows:

$$\begin{aligned}
& c_1(\mu_{1,1}^\sigma, \dots | \langle \rho_{\sigma,1}, \dots, \rho_{\sigma,m_\sigma} \rangle, \dots, \langle \rho_{14,1}, \dots, \rho_{14,m_{14}} \rangle) \\
& = w_1 \cdot \sum_{k \in \{1, \dots, \nu\}} \max\{h_k - 36, 0\} / (\nu \cdot 48) + w_2 \cdot \sum_{s \in \{\sigma, \dots, 14\}} \sum_{r \in \{1, \dots, \gamma_s\}} R_r^s / \sum_{s \in \{\sigma, \dots, 14\}} \sum_{i \in \{1, \dots, m_s\}} (2\delta_{0i}^s + \sum_{j \in \{1, \dots, n_{s,j}\}} \tau_{i,j}^s) \\
& \quad + \alpha \cdot \varepsilon_1 + \beta \cdot \varepsilon_2 + \beta \cdot \varepsilon_3 + \beta \cdot \varepsilon_4
\end{aligned} \tag{11}$$

where the cost of penalizing a hard constraint is  $\alpha$ , and the cost of penalizing a hard constraint is  $\beta$ . Fig. 7 shows the boxplots of costs of the experimental results using the proposed MHS algorithm for the city problem instance under different combinations of  $\alpha$  and  $\beta$ . From Fig. 7(a) (i.e., when  $\beta = 10$ ), the results when  $\alpha = 1,000$  and  $\alpha = 10,000$  do not have much difference, but the result when  $\alpha = 100$  performs worse, because the cost of 10 violations of soft constraints is equal to the cost of violating a hard constraint in this case, so that hard constraints may not be addressed by the algorithm. Similarly, the result when  $\alpha = 1,000$  and  $\beta = 100$  in Fig. 7(b) (i.e., the rightmost box) performs worse. Although the result when  $\beta = 1$  (i.e., the leftmost box in Fig. 7(b)) has the lowest cost, its performance is similar to the result when  $\beta = 10$  (i.e., the middle box in Fig. 7(b)) in terms of number of soft constraints. Therefore, we set  $\alpha = 1,000$  and  $\beta = 10$  in Equation (9).

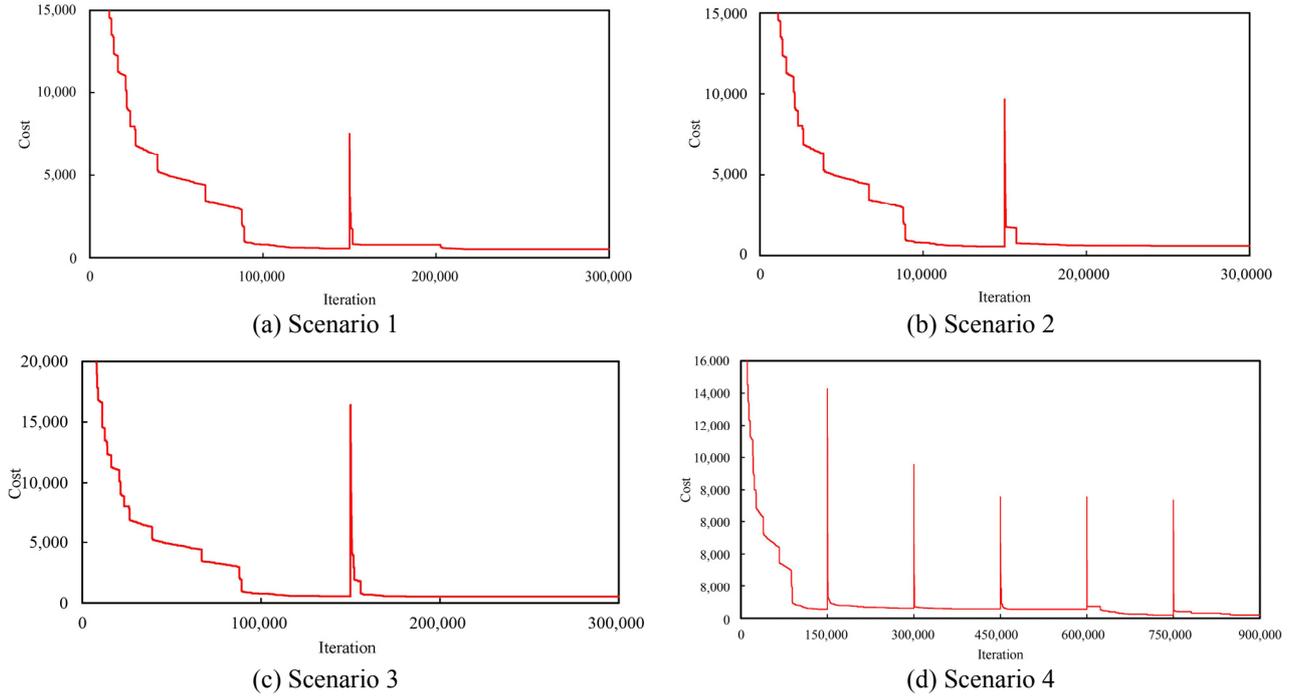


**Fig. 7.** Boxplots of costs of the experimental results using the proposed MHS algorithm for the city problem instance under (a) three different values of  $\alpha$  when  $\beta = 10$ , and (b) three different values of  $\beta$  when  $\alpha = 1,000$ .

### 5.3. Experimental analysis for the second model

This section further analyzes the experimental results for the second model, i.e., to check whether the proposed MHS algorithm can adjust the roster to adapt to occurrence of certain sudden incident, so that cost  $c_2$  is minimized. Five experimental scenarios are considered as follows:

- 1) *Scenario 1 (a nurse suddenly requests for a leave)*: This scenario considers that after a part of the roster for the first model has been finished, certain nurse suddenly requests to be absent in one of the remaining time slots. Our simulation selects a random number of time slots that have been finished and a random nurse to be absent in a random unfinished time slot. Let the number of iterations in the proposed MHS algorithm be 150,000. In the simulation results in Fig. 8(a), a roster for the first model is generated in the first 15,000 iterations (in which the cost gradually decreases from a high level to a low level). Then, suppose that Scenario 1 occurs at the 15,001st iteration, so that the cost rises to a much high level, in which some constraints are violated and the difference between the original and the new rosters is large. Then, the algorithm improves the cost back to a low level from the 15,001st iteration to the last iteration. Hence, it is concluded that the proposed MHS algorithm has the ability to adapt to this scenario.

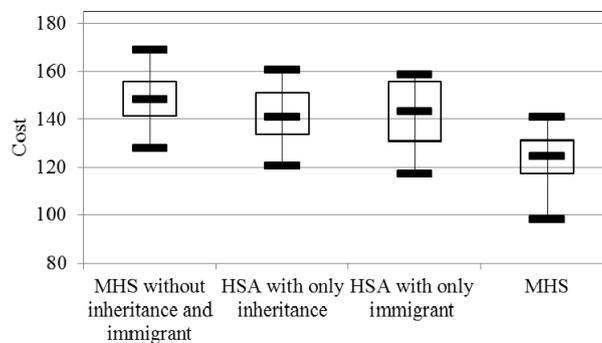


**Fig. 8.** Experimental results for rostering in four scenarios.

- 2) *Scenario 2 (a patient suddenly changes the time slot to be treated)*: This scenario considers that after a part of the roster for the first model has been finished, certain patient that is not served yet suddenly requests to change to be served in a different time slot. We set the simulation setting for checking whether this scenario can be addressed to be similar to Scenario 1. The simulation result is shown in Fig. 8(b). From this figure, the proposed MHS algorithm is shown to be able to adapt to this scenario.
- 3) *Scenario 3 (a patient suddenly requests for a leave)*: This scenario considers that after a part of the roster for the first model has been finished, certain patient in the remaining time slots suddenly requests to be absent or to quit the HHC service. Similarly, the simulation result is shown in Fig. 8(c), which confirms the adaption of the proposed MHS algorithm to this scenario.
- 4) *Scenario 4 (combination of the above three scenarios)*: Consider a dynamic scenario in which the above three scenarios occur randomly in one week. The simulation in Fig. 8(d) considers that one of the above three scenarios occurs in each of 150,000 iterations. Specifically, at the

150,001st iteration, time slots 1 and 2 have been finished, but Scenario 3 occurs. At the 300,001st iteration, time slots 3 and 4 have been finished, but Scenario 1 occurs. Similar settings are applied in the other iterations. The simulation result in Fig. 8(c) confirms the adaptation of the proposed MHS algorithm to consecutive occurrence of the above three scenarios.

- 5) To verify whether inheritance and immigrant schemes assist the proposed MHS algorithm in handling dynamic optimization problems, and to analyze the performance of these two schemes, this work experimentally compares four versions of the MHS algorithm: the MHS algorithm without inheritance and immigrant schemes, the MHS algorithm with only inheritance scheme, the MHS algorithm with only immigrant scheme, and the proposed MHS algorithm with both inheritance and immigrant schemes (Fig. 9). Note that for fairness of comparison, the initial input solutions of four version in this experiment for the second rostering problem model are the same (i.e., they are the results of the MHS algorithm in Fig. 6(b)). From Fig. 9, the proposed MHS algorithm with inheritance and immigrant schemes performs best, and both the two schemes have effects to reduce the cost.



**Fig. 9.** Boxplots of costs of the experimental results using the MHS algorithm without inheritance and immigrant schemes, the MHS algorithm with only inheritance scheme, the MHS algorithm with only immigrant scheme, and the proposed MHS algorithm with both inheritance and immigrant schemes.

Furthermore, every time when there is a sudden event, if the duty roster is not adjusted, the cost rises to a much high level. Especially, the cost in all scenarios exceeds 1,000 (Fig. 8), meaning that the roster violates a lot of hard constraints, i.e., this solution is infeasible. Even if it is feasible, it represents that the roster violates thousands of soft constraints, so that nurses are much dissatisfied

with such a roster. The second model in this work additionally considers minimizing the differences between the new nurse roster and the original nurse roster in which some work shifts have been finished. Hence, the proposed MHS algorithm reduces the number of hard and soft constraints when have too much change on the nurse roster. Sometimes, the MHS algorithm can find a better nurse roster than the one before change, e.g., the process between 600,000th and 750,000th iterations in Fig. 8(d). Therefore, the second model that considers rostering, routing, and rostering is necessary.

## 6. Conclusion

This work has proposed an MHS algorithm with genetic, saturation, inheritance, and immigrant schemes for the problem that jointly considers rostering, routing, and rostering for home health care services ( $R^3$ HHCS). To the best of our knowledge, all the previous works on the HHCP separately addressed the NRP and the VRPTW of the HHCP (e.g., Nickel et al. (2012)). However, the optimal solutions respectively for the NRP and the VRPTW may be in conflict with each other in many cases. Hence, this work proposed an approach that concurrently optimizes both the NRP and the VRPTW of the HHCP. Different from the previous works on the HHCP (e.g., Nickel et al. (2012)) that assigned dummy nurses to serve some tasks, the solution generated by the proposed algorithm guaranteed each task of patients to be assigned to a nurse, if the number of nurses are enough. Nurse preferences and nurse rostering constraints in the HHCP were considered to increase the total nurse satisfaction.

The  $R^3$ HHCS included two problem models. The first model aimed to minimize the total nurse overtime cost and the total vehicle routing cost, under nurse qualification constraints, the working constraints defined by laws and contracts, nurse preference constraints, and vehicle routing constraints, so that nurses can be more satisfied with the roster, and both occupational injury and cost could be reduced. The second model considered that occurrence of a sudden incident leads to rostering, and aimed to minimize the difference between the original and new rosters in which a part of the time slots has been finished and to minimize the total cost at the same time.

The MHS algorithm extended the HSA with genetic, saturation, inheritance, and immigrant schemes, in which either the conventional HSA operators or the GA operators are executed

according to a saturation value that changes iteratively. By the saturation scheme, repeated solutions could be avoided so as to increase the solution searching efficiency. The experimental and statistical analysis for the first model showed that the proposed MHS algorithm not only can generate good-quality solutions but also is executed more efficiently than the HSA. For the second model, three scenarios for sudden incidents were considered, and the simulation results showed that the proposed MHS algorithm can adapt to occurrence of these sudden incidents to produce good-quality rosters.

In the future, a line of the research is to consider more practical factors or constraints in HHC services, e.g., vehicle breakdown, routing road damage, traffic congestion, nurse skill training, policy making, and service region scale.

### **Acknowledgements**

The authors thank the anonymous referees for comments that improved the content as well as the presentation of this paper. This work has been supported in part by the Ministry of Science and Technology, Taiwan, under Grants MOST 105-2628-E-009-002-MY3 and MOST 104-2221-E-009-101.

### **References**

- Akjiratikarla, C., Yenradeea, P., & Drakeb, P. R. (2007). PSO-based algorithm for home care worker scheduling in the UK. *Computers & Operations Research*, 33(4), 559–583.
- Allaoua, H., Borne, S., L'etocart, L., & Calvo, R. W. (2013). A matheuristic approach for solving a home health care problem. *Electronic Notes in Discrete Mathematics*, 41, 471–478.
- Bagheri, M., Devin, A. G., & Izanloo, A. (2016). An application of stochastic programming method for nurse scheduling problem in real world hospital. *Computers & Industrial Engineering*, 96, 192–200.
- Bennett-Milburn, A., & Spicer, J. (2013). Multi-objective home health nurse routing with remote monitoring devices. *International Journal of Planning and Scheduling*, 1(4), 242–63.
- Bertels, S., & Fahle, T. (2006). A hybrid setup for a hybrid scenario: Combining heuristics for the home health care problem. *Computers and Operations Research*, 33(10), 2866–2890.
- 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.
- Cappanera, P., & Scutellà, M. G. (2014). Joint assignment, scheduling, and routing models to home care optimization: a pattern-based approach. *Transportation Science*, 49(4), 830–852.

- Cheng, E., & Rich, J. L. (1998) A Home Health Care Routing and Scheduling Problem. Technical Report, Rice University.
- Cheng, H. & Yang, S. (2010) Genetic algorithms with immigrants schemes for dynamic multicast problems in mobile ad hoc networks. *Engineering Applications of Artificial Intelligence*, 23(5), 806–819.
- Clark, P. A., Leddy, K., Drain, M., & Kaldenberg, D. (2007). State nursing shortages and patient satisfaction: More RNs–better patient experiences. *Journal of Nursing Care Quality*, 22(2), 119–127.
- Cordeau, J.-F., Desaulniers, G., Desrosiers, J., Solomon, M.M., & Soumis, F. (2002). The Vehicle Routing Problem. In P. Oth and D. Vigo (Eds.), *SIAM Monographs on Discrete Mathematics and Applications*, pp. 176–213.
- Dohn, A., Rasmussen, M. S., & Larsen, J. (2011). The vehicle routing problem with time windows and temporal dependencies. *Networks*, 58(4), 273–289.
- Eveborn, P., Flisberg, P., & Rönnqvist, M. (2006). Laps Care—An operational system for staff planning of home care. *European Journal of Operational Research*, 171(3), 962–976.
- Fikar, C., & Hirsch, P. (2015). A matheuristic for routing real-world home service transport systems facilitating walking. *Journal of Cleaner Production*, 105, 300–310.
- Fikar, C., & Hirsch, P. (2017). Home health care routing and scheduling: A review. *Computers & Operations Research*, 77, 86–95.
- Frosolini, M., Braglia, M., & Zammori, F. (2010). A modified harmony search algorithm for the multi-objective flowshop scheduling problem with due dates. *International Journal of Production Research*, 49, 5957–5985.
- Geem, Z. W., Kim, J. H., & Loganathan, G. V. (2001). A new heuristic optimization algorithm: harmony search. *Simulation*, 76(2), 60–68.
- Goldberg, D. E., & Lingle, R. (1985). Alleles, Loci and the traveling salesman problem. In: *Proceedings of the First International Conference on Genetic Algorithms and Their Applications*, ACM Press, pp. 154–159.
- Hadwan, M., Ayob, M., Sabar, N. R., & Qu, R. (2013). A harmony search algorithm for nurse rostering problems. *Information Sciences*, 33(1), 26-140.
- Hallah, R. M., & Alkhabbaz, A. (2013). Scheduling of nurses: A case work of a Kuwaiti health care unit. *Operations Research for Health Care*, 2(1-2), 1–19.
- Holland, J. H. (1992). *Adaptation in Natural and Artificial Systems: An Introductory Analysis with Applications to Biology*. Control and Artificial Intelligence, The MIT Press, Bradford, UK.
- Ioria, M., & Riera-Ledesma, J. (2015). Exact algorithms for the double vehicle routing problem with multiple stacks. *Computers & Operations Research*, 63, 83–101.

- Kallehauge, B., Larsen, J., Madsen, O. B., & Solomon, M. (2005). The vehicle routing problem with time windows. In G. Desaulniers, J. Desrosiers, M.M. Solomon (Eds.), *Column Generation, GERAD 25th Anniversary Series*, Springer, pp. 67–98.
- Lee, K. S., & Geem, Z. W. (2004). A new structural optimization method based on the harmony search algorithm. *Journal of Computers and Structures*, 82, 781–798.
- Lessel, C. R. (2007). *Ruteplanlægning i hjemmeplejen*. Master's thesis, Informatics and Mathematical Modelling, Technical University of Denmark.
- Liu, R., Xie, X., Augusto, V., & Rodriguez, C. (2013). Heuristic algorithms for a vehicle routing problem with simultaneous delivery and pickup and time windows in home health care. *European Journal of Operational Research*, 230(3), 475–486.
- Marcelli, E., Scalambra, F., Cercenelli, L., & Plicchi, G. (2007). A new hermetic antenna for wireless transmission systems of implantable medical devices. *Medical Engineering & Physics*, 29(1), 140–147.
- Maya Duque, P. A., Castroa, M., Sörensen, K., & Goosa, P. (2015). Home care service planning. The case of Landelijke Thuiszorg. *European Journal of Operational Research*, 243(1), 292–301.
- Moz, M., & Pato, M. V. (2004). Solving the problem of rostering nurse schedules with hard constraints: new multicommodity flow models. *Annals of Operations Research*, 128, 179–197.
- Moz, M., & Pato, M. V. (2007). A genetic algorithm approach to a nurse rostering problem, *Computers & Operations Research*, 34(3), 667–691.
- Nickel, S., Schroder, M., & Steeg, J. (2012). Mid-term and short-term planning support for home health care services. *European Journal of Operational Research*, 219, 574–587.
- Rasmussen, M. S., Justesen, T., Dohn, A., & Larsen, J. (2012). The home care crew scheduling problem: preference-based visit clustering and temporal dependencies. *European Journal of Operational Research*, 219(3), 598–610.
- Soysala, M., Bloemhof-Ruwaard, J. M., & Bektas, T. (2015). The time-dependent two-echelon capacitated vehicle routing problem with environmental considerations. *International Journal of Production Economics*, 164, 366–378.
- Steeg, J. (2008). *Mathematical Models and Algorithms for Home Health Care Services*. Dissertation, Technische Universität Kaiserslautern.
- Thomsen, K. (2006). *Optimization on Home Care*. Master's thesis, Informatics and Mathematical Modelling, Technical University of Denmark.
- Turky, A. M., & Abdullah, S. (2014). A multi-population harmony search algorithm with external archive for dynamic optimization problems. *Information Sciences*, 272, 84-95.
- Turky, A. M., Abdullah, S., & Sabar, N. R. (2014). A hybrid harmony search algorithm for solving dynamic optimisation problems. *Procedia Computer Science*, 29, 1926-1936.

- Wang, Z., Li, Y., & Hu, X. (2015). A heuristic approach and a tabu search for the heterogeneous multi-type fleet vehicle routing problem with time windows and an incompatible loading constraint. *Computers & Industrial Engineering*, 89, 162–176.
- Yalçındağ, S., Cappanera, P., Scutellà, M. G., & Şahin, E. (2016). A. Matta, Pattern-based decompositions for human resource planning in home health care services. *Computers & Operations Research*, 73, 12–26.
- Yassen, E. T., Ayob, M., Nazri, M. Z. A., & Sabar, N. R. (2015a). A Hybrid Meta-Heuristic Algorithm for Vehicle Routing Problem with Time Windows. *International Journal on Artificial Intelligence Tools*, 24(6), 1550021.
- Yassen, E. T., Ayob, M., Nazri, M. Z. A., & Sabar, N. R. (2015b). Meta-harmony search algorithm for the vehicle routing problem with time windows. *Information Sciences*, 325, 140–158.
- Zammori, F., Braglia, M., & Castellano, D. (2014). Harmony search algorithm for single-machine scheduling problem with planned maintenance. *Computers & Industrial Engineering*, 76, 333–346.