Abstract: In this paper, we introduce two mathematical programming models of the resident scheduling problem (RSP). These were designed to solve the RSP in rotation and night-shift scheduling of residents at a hospital in Japan. Automatic scheduling is designed to support management in challenges such as increasing the satisfaction of residents and reducing the scheduling workload. To achieve this, we placed weighted penalties on soft constraints and used objective functions to automatically generate the schedules. The results improved the management of resident scheduling, and comparisons with schedules drawn up by hand confirmed the superiority of those produced by the systems. Following some fine-tuning based on user feedback, our night-shift scheduling system was put into use at Aichi Medical University Hospital in January 2016.

Key words: Resident scheduling problem, Mathematical programming problem, Analytic hierarchy process

1 INTRODUCTION

Residents are medical doctors who have graduated from medical school and are engaged in residency training at a hospital [3]. The number of residents in Japan has been increasing, with 8,244 residents, the largest number ever, in training at 1,018 teaching hospitals and university hospitals in 2015 [1]. From April 1, 2004, the Japanese government has required residents to complete a two-year training program [4],[5]. These residents are assigned to a new department each month, and thus rotate between many departments throughout the training period.

Many rural and small- to medium-sized urban hospitals in Japan are facing a shortage of residents [2]. As they receive fewer clinical cases than large urban hospitals, the training environment is less attractive to potential residents who wish to acquire a wider range of clinical skills. This creates competition between hospitals to attract residents, which requires them to continuously improve their training environment. Part of this challenge, which is faced by all hospitals, is to find a way of offering residents a rigorous training program that also affords a sensible work/life balance.

Most residents wish to train in their future specialization, and can request that they be located in the appropriate department. However, not all these requests can be satisfied, as the hospital must consider the number of physician trainers available and the capacity of the department. Having their request denied can be a cause of stress for residents, on top of the stress of working long hours, including night shifts [5]. To minimize these stresses, the hospital must draw up schedules that take the preferences and workload of the resident into account. The main goal of rotation scheduling is to determine in which department each resident will be placed each month. Scheduling also determines when and how often each resident is assigned to a night shift, weekend shift, or holiday shift. This so-called resident scheduling problem (RSP) has recently been addressed using mathematical programming.

Aichi Medical University Hospital is a large hospital in Aichi Prefecture, Japan. It has approximately 900 beds among its 30 departments, and
employs around 30 residents each year. Resident rotation and shift scheduling are currently handled manually by staff of the Postgraduate Clinical Training Center. This is a time-consuming process because of the large number of constraints that must be taken into account when creating a schedule. It typically takes three days to create a yearly rotation schedule and 24 hours to create the monthly shift schedule. The schedules may not always be sufficiently accurate or comprehensive because all requirements are checked manually.

In the rotation schedule, a resident may be allocated to a clinical area in which s/he does not require training. In a properly balanced rotation schedule, each resident should rotate only among those departments which s/he has requested. In practice, each department can only accommodate a certain number of residents, and the rotation schedule must also balance resident allocation with the number of physician trainers available. Accordingly, satisfying both of these requirements simultaneously is very challenging. Ideally, residents should rotate between departments in a pattern that satisfies as many constraints as possible. A further complication is that the number of residents often changes immediately before the schedule is launched, as the results of the National Medical Practitioners Qualifying Examination are announced in mid-March, and not all of the prospective residents pass. This requires immediate changes to be made to the rotation schedule.

Scheduling often produces imbalances in the allocation of night shifts, weekend shifts, and holiday shifts. As schedules are created on a month-by-month basis, the staff devising them may not take the previous month’s allocation into consideration. Residents may then re-distribute their allocation through personal negotiation.

In this study, we implemented two systems for addressing the RSP at Aichi Medical University Hospital: a rotation scheduling system for residents (RSSR) and a night-shift scheduling system for residents (NSSR). These systems successfully reduced the load placed on staff. We formulated each as a mathematical programming problem, and then implemented the systems in Excel using CPLEX optimization software. Resident rotation scheduling was formulated as an integer programming problem, and night-shift scheduling as a weighted constraint satisfaction problem in which the weights were set using the analytic hierarchy process (AHP). As the NSSR requires the one-month rotation schedule created by the RSSR, we developed the latter in such a way that the NSSR has automatic access to its schedules. The NSSR has been under trial application at Aichi Medical University Hospital since January 2016, and the RSSR has now been added to this trial.

A number of previous studies have analyzed the RSP. Resident training in other countries has been shown to involve long working hours [7,8]. At one university hospital in Canada, nearly half the residents were working more than 80 hours per week, and one in six more than 100 hours per week. Different countries have developed a range of systems for resident training. Hannah and Pinar [13] addressed physician scheduling in a pediatric intensive care unit using a mixed-integer programming model, and proposed a heuristic method for maximizing the continuity of care. Topaloglu and Ozkarahan [11] managed the day and night schedules of residents while balancing the constraints of resident preference, educational value, and patient safety. They treated this as a mixed-integer programming problem, solved it using real hospital data, and tested the validity of the model. Hannah and Pinar [14] solved the rotation-scheduling and day/night-shift scheduling problems by formulating both as integer programming problems. They demonstrated an improvement in working efficiency by automating schedule creation.

Despite the progress that has been made in other countries, mathematical programming has not yet been applied to the RSP in Japan, and automatic scheduling systems have not yet been developed. The country-specific approaches developed elsewhere cannot be used in Japan because the country’s residency system is significantly different. For example, residency training lasts for 12 months in most countries, but is 24 months in Japan. In the U.S., the Accreditation Council for Graduate Medical Education limits the working hours of residents to a maximum of 80 hours per week. Japan’s Labor Standards Act sets a limit of 40 hours per
week, but if more than 40 hours are worked, they are paid the salaries for overtime or holiday pay. In practice, therefore, Japan sets no upper limit on working hours. As there are relatively few doctors per patient, residents often have to work beyond their official hours.

The models for rotation scheduling and night-shift scheduling introduced in this study were applied to real problems, and this paper reports both the design and implementation of the systems. The remainder of the paper is organized as follows. Section 2 gives a detailed introduction of RSSR implementation and the formulation of rotation scheduling as an integer programming problem. It also discusses the application of the method at Aichi Medical University Hospital. Section 3 introduces the NSSR, shows how it formulates resident night-shift scheduling as a 0-1 integer programming problem, and discusses the use of AHP to obtain the appropriate weightings. Again, the results of applying the method at Aichi Medical University Hospital are presented. Section 4 concludes the paper and gives proposals for future research.

2 ROTATION SCHEDULING

2.1 RSSR Implementation

The RSSR assigns residents to departments, and was designed in such a way that the schedule can be easily modified when the constraints change unexpectedly, providing a solution that maximizes the satisfaction of the residents. It allows residents to be assigned to departments in as equal numbers as possible within each monthly schedule. RSSR also creates a two-year rotation schedule for first-year residents. The input data are the names of the residents, rotation of the residents in April and May of the first year and May to March of the second year, and the preferences of the residents on assignment to the five internal medicine departments. The output data are the rotation of the residents from June of the first year to April of the second year. The goal was to achieve full implementation of the system at Aichi Medical University Hospital. Figure 1 shows a schematic of the RSSR interface screen. We designed the RSSR so that the rotation schedule generated by the CPLEX optimization software was presented via an Excel interface.

The “Basic” button shown in Fig. 1 is used to enter the number of residents and their names. The “Department” button is used to set the lower and upper monthly capacity limits of each department. The “Demand” button is used to set the internal medicine departments in which each resident should receive training. Clicking the “Rotate” button creates and displays the rotation schedule that solves the scheduling problem using CPLEX.

2.2 Model of the Rotation Scheduling Problem for Residents

This section introduces a model of the rotation-scheduling problem, in which residents must be allocated to the most suitable department. The staff schedulers at Aichi Medical University Hospital attempt to create the schedule that best satisfies the wishes of the residents, while balancing the number of residents assigned to different departments. It was formulated as a mixed-integer programming problem as follows.

**Notation**

**Index sets**

- \( R \): set of residents
- \( M \): set of 13 months from April to the following April
- \( M_a \): set of months in April and May
- \( M_{ac} \): set of months from April to the following February
- \( M_k \): set of months from April to the following March
- \( M_2 \): set of months from May to the following March
- \( D \): set of departments
- \( D_{int} \): set of internal medicine departments, \( D_{int} \subset D \)
- \( D_j \): set of other departments, \( D_j \subset D, D_j \neq D_{int} \)
- \( D_a \): set of anesthesiology department, \( D_a \subset D_j \)
- \( D_e \): set of emergency department, \( D_e \subset D_j \)
- \( D_{sur} \): set of general surgery department, \( D_{sur} \subset D_j \)
- \( D_{ss} \): set of specialized surgeries department, \( D_{ss} \subset D_j \)
- \( D_p \): set of pediatric departments, \( D_p \subset D_j \)
Parameters

\( n_{rd} \): 1 if resident \( r \) requests the internal medicine department \( d \), otherwise 0
\( n_{rd} \): number of months that resident \( r \) needs to train in department \( d \)
\( d_{md}^{-} \): lower capacity limit of department \( d \) in month \( m \)
\( d_{md}^{+} \): upper capacity limit of department \( d \) in month \( m \)
\( mT \): final month in the scheduling period
\( tr_{rd} \): 1 if resident \( r \) trains in department \( d \) in month \( m \), otherwise 0 (initial value in April and May)
\( \alpha, \beta, \gamma, \delta \): weights that determine the priority of each term in the objective function

Variables

\( x_{rdm} \): 1 if resident \( r \) is assigned to department \( d \) in month \( m \), otherwise 0
\( dp_{rd} \): penalty for assigning resident \( r \) to a department other than department \( d \) against her/his wishes
\( aep_{rm} \): penalty for non-assigning resident \( r \) to the anesthesiology and emergency departments for four consecutive months \( ms \)
\( up_{rdm} \): penalty for allocating a different internal medicine department to resident \( r \) for consecutive months \( ms \)
\( ep_{rm} \): penalty for not assigning the emergency department to resident \( r \) for two consecutive months \( ms \)

Formulation

\[
\begin{align*}
& \text{minimize} & \alpha \sum_{r \in R} \sum_{d \in D_{nt}} dp_{rd} \\
& & + \beta \sum_{r \in R} \sum_{m \in M_{ae}} aep_{rm} \\
& & + \gamma \sum_{r \in R} \sum_{m \in M_{ae}} \sum_{d \in D_{nt}} up_{rdm} \\
& & + \delta \sum_{r \in R} \sum_{m \in M_{2}} ep_{rm} \\
& \text{s.t.} & \sum_{m \in M} x_{rdm} \geq n_{rd} - dp_{rd}, & r \in R, d \in D_{nt} \quad (2) \\
& & x_{rdm} - x_{r,m+2,d^{'}} \leq aep_{rm}, & r \in R, m \in M_{ae}, d \in D_{a}, d^{'} \in D_{e} \quad (3) \\
& & x_{rdm} - x_{r,m+1,d} \leq up_{rdm}, & r \in R, m \in M_{2}, d \in D_{nt} \quad (4) \\
& & x_{rdm} - x_{r,m+1,d} \leq ep_{rm}, & r \in R, m \in M_{2}, d \in D_{e} \quad (5) \\
& & x_{rdm} = tr_{rdm}, & r \in R, m \in M_{a}, d \in D \quad (6) \\
& & x_{rdm} \leq x_{r,m-1,d^{'}} + x_{r,m+1,d^{'}} & r \in R, m \in M_{2}, d \in D_{sur}, d^{'} \in D_{ss} \quad (7) \\
& & x_{r,mT,d} \leq x_{r,mT-1,d^{'}} & r \in R, d \in D_{sur}, d^{'} \in D_{ss} \quad (8) \\
& & d_{md}^{-} \leq \sum_{r \in R} x_{rdm} \leq d_{md}^{+} & m \in M, d \in D \quad (9) \\
& & \sum_{m \in M} x_{rdm} = n_{rd}, & r \in R, d \in D_{j} \quad (10)
\end{align*}
\]
\[
\sum_{d \in D} x_{rmd} \leq 1, \quad r \in R, m \in M
\]  \quad (11)

\[
d_{prd} \in \{0, 1\}, \quad r \in R, d \in D_{int}
\]  \quad (12)

\[
u_{rmd} \in \{0, 1\}, \quad r \in R, m \in M_{s}, d \in D
\]  \quad (13)

\[
e_{prm} \in \{0, 1\}, \quad r \in R, m \in M_{a}
\]  \quad (14)

\[
ae_{prm} \in \{0, 1\}, \quad r \in R, m \in M_{ae}
\]  \quad (15)

\[
x_{rmd} \in \{0, 1\}, \quad r \in R, m \in M, d \in D
\]  \quad (16)

In the formulation shown above, objective function (1) is the minimization of the sum of the four penalties. The first term is the sum of the penalty \(d_{prd}\), defined in (2) as the penalty for assigning resident \(r\) to department \(d\) against their wishes. The second term is the sum of the penalty \(\nu_{rmd}\), defined in (3) as the penalty for not assigning resident \(r\) to the anesthesiology and emergency departments for four consecutive months. The third term is the sum of the penalty \(e_{prm}\), defined in (4) as the penalty for assigning resident \(r\) to a different internal medicine department for more than one month consecutively. The fourth term is the sum of the penalty \(ae_{prm}\), defined in (5) as the penalty for not assigning resident \(r\) to the emergency department for two consecutive months.

Constraint (2) captures the assumption that resident \(r\) should, where possible, be assigned to his/her requested department \(d\). Constraint (3) means that resident \(r\) should, where possible, be assigned to the anesthesiology and emergency departments for four consecutive months. Constraint (4) assumes that resident \(r\) should, where possible, be assigned to the same department for consecutive months. Constraint (5) means that resident \(r\) should, where possible, be assigned to the emergency department for two consecutive months. It is desirable that resident \(r\) rotates from the anesthesiology department to the emergency department. Constraint (6) assumes that the department to which each resident is assigned is determined in advance. Constraints (7) and (8) assume that all residents should be assigned to the general surgery and specialized surgeries departments for two consecutive months. Constraint (9) is the upper and lower monthly capacity limit of each department.

Constraint (10) sets the number of months that resident \(r\) should train in department \(d\). Constraint (11) means that each resident should be assigned to a single department within any month. Constraints (12), (13), (14), (15), and (16) are binary constraints.

We can classify these constraints into three groups: constraints (7), (8), (9) and (12) arise from the wider medical system, constraints (3), (4) and (10) are specific to the systems of Aichi Medical University Hospital, and constraints (2), (5), (6) and (11) reflect the goals of the staff scheduler.

2.3 Results

We created rotation schedules for 2015 using real data from Aichi Medical University Hospital. Fig.
Table 1  Number of residents each month in sections of the 2015 rotation schedules that were drawn up manually: I, internal medicine; PC, primary care; S, general surgery; SS, specialized surgeries; A, anesthesiology; E, emergency; P, pediatrics.

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>I</td>
<td>10</td>
<td>10</td>
<td>9</td>
<td>9</td>
<td>9</td>
<td>9</td>
<td>9</td>
<td>9</td>
<td>11</td>
<td>6</td>
<td>9</td>
<td>8</td>
<td>15</td>
</tr>
<tr>
<td>S</td>
<td>5</td>
<td>5</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>0</td>
</tr>
<tr>
<td>SS</td>
<td>0</td>
<td>0</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>0</td>
</tr>
<tr>
<td>A</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>3</td>
<td>5</td>
<td>3</td>
<td>4</td>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td>E</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>3</td>
<td>5</td>
<td>3</td>
<td>4</td>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td>P</td>
<td>0</td>
<td>0</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>2</td>
</tr>
</tbody>
</table>

Figure 2a and 2b show extracts from the 2015 rotation schedules drawn up manually and using the RSSR. It is clear from Fig. 2a that, in the manual rotation schedule, other departments were fit between the training blocks allocated to the anesthesiology (A) and emergency (E) departments. The RSSR (Fig. 2b) allocated the anesthesiology and emergency departments as consecutive blocks to all but four of the residents.

Tables 1 and 2 list the number of residents each month in sections of the 2015 rotation schedules that were drawn up manually and using the RSSR, respectively. Table 1 shows that the maximum number of residents assigned to a department was fifteen, and the minimum was six, giving a difference of nine. Table 2 shows that the maximum number of residents assigned to a department was fourteen and the minimum was seven, giving a difference of seven. The proposed system allocated residents in such a way as to reduce the difference between the maximum and minimum.

Scheduling was done on a computer running Microsoft Windows 7, with an Intel Core i5-2450M CPU and 4 GB of RAM. The parameter values in the cases examined were as follows: number of residents $|R| = 25$; number of months $|M| = 13$; and number of departments $|D| = 14$. A CPU time of 40 s was required to solve the mixed-integer programming problem using CPLEX Optimization Studio Version 12.5.1. The problem had 7,476 variables and 8,614 constraints, and the optimality gap was 1%. An upper limit on computation time of 2 min was set.

3  NIGHT-SHIFT SCHEDULING FOR RESIDENTS

3.1 NSSR Implementation

We implemented the NSSR using VBA (Visual Basic for Applications) in Excel. Figure 3 shows the NSSR user interface. Schedules can be created automatically using simple operations.

To begin with, the user clicks the cell marked “November 2015” shown in Fig. 3 and selects the relevant month from a drop-down menu. Next, the user clicks the “Set the month” button, bringing up a calendar of the selected month and a list of the departments that were assigned to that month by the RSSR.

The button marked “Set the resident’s off-shifts” is used to set the shifts for which each resident should not be assigned night duty. Clicking this button automatically enters “X” in the appropriate cells of each resident’s off-shifts. These off-shifts form part of the program that implements NSSR, as they do not change from year to year. These off-shifts are determined by the training department so that, for example, a resident assigned to the circulation department cannot work the Sunday night shift that precedes the regular Monday morning meeting. To account for individual plans, such as an off-duty shift requested by a resident or attendance at an academic conference, the scheduler enters these using the buttons marked “X”, “echo”, “AED”, “ICT”, “ICLS” and “Other”. These are shown in Fig. 3, omitting buttons that display other types of off-shift. The residents have reported scheduler the individual plans before the creation date of night-shift schedule. To the right of these, the buttons marked “C” and “E” set the shifts
Table 2  Number of residents each month in sections of the 2015 rotation schedules using the RSSR: I, internal medicine; PC, primary care; S, general surgery; SS, specialized surgeries; A, anesthesiology; E, emergency; P, pediatrics.

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>10</td>
<td>10</td>
<td>9</td>
<td>9</td>
<td>11</td>
<td>10</td>
<td>8</td>
<td>7</td>
<td>7</td>
<td>11</td>
<td>9</td>
<td>14</td>
</tr>
<tr>
<td>S</td>
<td>5</td>
<td>5</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>SS</td>
<td>0</td>
<td>0</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>0</td>
</tr>
<tr>
<td>A</td>
<td>4</td>
<td>4</td>
<td>3</td>
<td>4</td>
<td>3</td>
<td>3</td>
<td>5</td>
<td>6</td>
<td>4</td>
<td>5</td>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td>E</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>3</td>
<td>4</td>
<td>3</td>
<td>3</td>
<td>5</td>
<td>5</td>
<td>3</td>
<td>5</td>
</tr>
<tr>
<td>P</td>
<td>0</td>
<td>0</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
</tr>
</tbody>
</table>

Night-shift scheduling system for residents (November 2015)

![NSSR user interface](image)

Fig. 3  NSSR user interface.

for which each resident should be assigned night duty (again, we have omitted other buttons that display other types of on-shift). Here, “C” means critical care center shift and “E” means intensive care unit (ICU) shift. The button marked “Set the data” prepares the necessary data for scheduling. For instance, the button inputs each resident’s off-shifts and on-shifts. The button marked “Check for contradictions” prompts the NSSR to check for any false entries made by the operators, and confirms that there are a correct number of residents and blank cells. The button marked “Scheduling” solves the resident night-shift scheduling problem, which is formulated as a 0-1 integer programming problem, as discussed in the next section. The button marked “Output” displays the schedule generated by the “Scheduling” button. A button marked “Back to set the resident’s off-shifts” appears before the “Scheduling” button is clicked, and the button marked “?” shows the manual for operating the NSSR.

3.2 Model of the Night-shift Scheduling Problem for Residents

We next introduce a model of night-shift scheduling that allocates residents to the most suitable shifts while satisfying as many of the constraints as possible. The staff schedulers of Aichi Medical University Hospital need to create a schedule that gives each resident his/her preferred off-shifts, while balancing the number of night shifts and days of the week. The target shifts are weekday night shifts, and holiday day and night shifts. We assigned three types of work (critical care, ICU, and pediatric) to each shift. The night-shift scheduling problem was formulated as a 0-1 integer programming problem, as follows.
Parameters

Index sets

$R$: set of residents
$D$: set of departments
$K$: set of working forms.

$k = 1$ is critical care center, $k = 2$ is ICU of the set of working forms $K$.

$T$: set of shifts
$M$: set of days of the week.

$m = 1$ is Monday, $m = 2$ is Tuesday, $m = 3$ is Wednesday, $m = 4$ is Thursday, $m = 5$ is Friday, $m = 6$ is Saturday, $m = 7$ is Sunday of the set of days $M$.

$F$: set of first-year residents
$S$: set of second-year residents
$N$: set of night shifts, $N \subset T$
$Day$: set of day shifts, $Day \subset T$
$Res$: set of residents rotating in departments

except off-hospital, emergency, gynecology, anesthesiology
$L$: set of constraints

Notation

Variables

Formulation

\[
\begin{align*}
\text{minimize} & \quad \alpha \sum_{r \in Res} p_{r1} + \beta \sum_{r \in Res} p_{r2} \\
& \quad + \gamma \left( \sum_{r \in Res} p_{r3} + \sum_{r \in Res} p_{r4} \right) + \delta \sum_{r \in Res} p_{r5} \\
\text{s.t.} & \quad \sum_{k \in K} \sum_{t \in T} W_{rt}x_{rtk} \leq p_{r1}, \quad r \in Res \\
& \quad \sum_{t \in T} x_{rt1} + \sum_{t \in T} E_t x_{rt2} + 2 \sum_{t \in T} (1 - E_t)x_{rt2} \\
& \quad + \sum_{t \in T} q_{rt} + \sum_{t \in T} ped_{rt} \leq UB + p_{r2}, \quad r \in Res \\
& \quad \sum_{t \in T} (1 - E_t)(x_{rt1} + 2x_{rt2} + q_{rt} + ped_{rt}) \\
& \quad \leq HUB + p_{r3}, \quad r \in Res \\
& \quad \sum_{t \in T} (1 - E_t)(x_{rt1} + 2x_{rt2} + q_{rt} + ped_{rt}) \\
& \quad \geq HLB - p_{r4}, \quad r \in Res \\
& \quad \sum_{t \in T} \sum_{k \in K} G_{mt}(x_{rtk} + q_{rt} + ped_{rt}) \\
& \quad \leq 1 + p_{r5}, \quad r \in Res, m \in M \\
& \quad \sum_{r \in R} x_{rt1} + \sum_{r \in R} q_{rt} = 4, \quad t \in T \\
& \quad \sum_{r \in F} x_{rt1} + \sum_{r \in F} q_{rt} \geq FRL, \quad t \in T \\
& \quad \sum_{r \in F} x_{rt1} + \sum_{r \in F} q_{rt} \leq FRU, \quad t \in T \\
& \quad \sum_{r \in S} x_{rt1} + \sum_{r \in S} q_{rt} \geq SRL, \quad t \in T \\
& \quad \sum_{r \in S} x_{rt1} + \sum_{r \in S} q_{rt} \leq SRU, \quad t \in T
\end{align*}
\]
Parameters

Index sets

Notation

SRU: upper bound of required second-year residents

UB: lower bound of required first-year residents

Ane: lower bound of weekend and holiday shifts

M: set of working forms

T: 1 if shift

Res: 1 if resident

Day: 1 if off-shift of resident

Hrt: set of working forms

Wrt: 1 if resident is on duty in working form

Nrt: 1 if resident rotates to department

Krt: set of residents

p_rtl: penalty of constraint

\begin{align*}
\sum_{r \in R} x_{rt2} + \sum_{r \in R} ICU_{rt} &= 1, \quad t \in N \tag{28} \\
\sum_{r \in R} x_{rt2} &= 0, \quad t \in Day \tag{29} \\
\sum_{t \in T, k \in K} B_{rd} x_{rtk} &\leq 2, \quad r \in R, d = Ane \tag{30} \\
\sum_{k \in K} x_{rtk} + H_{rt} &\leq 1, \quad r \in R, t \in T \tag{31} \\
\sum_{k \in K, t \in T} x_{rtk} + \sum_{t \in T} q_{rt} + \sum_{t \in T} ped_{rt} &\geq LB, \quad r \in Res \tag{32} \\
G_{gt} \sum_{k \in K} x_{rtk} &\leq 1, \quad r \in R, t \in Day \tag{33} \\
G_{gt} \sum_{k \in K} x_{rtk} &\leq 1, \quad r \in R, t \in N \tag{34} \\
\sum_{t \in Day, k \in K} x_{rtk} &\leq daycount, \quad r \in R \tag{35} \\
\sum_{t \in T} x_{rt2} &\leq 1, \quad r \in R \tag{36} \\
\sum_{k \in K, t' = t} x_{rt'k} + \sum_{t' = t} q_{rt'} + \sum_{t' = t} ped_{rt'} &\leq 1, \quad r \in R, t \in T \tag{37} \\
\sum_{k \in K, t' = t} (1 - E_{t'}) x_{rt'k} &\leq 1, \quad r \in R, t \in T \tag{38} \\
p_{rt} &\geq 0, \quad r \in R, l \in L \tag{39} \\
x_{rtk} &\in \{0, 1\}, \quad r \in R, t \in T, k \in K \tag{40}
\end{align*}

In the above formulation, the objective function (17) is the minimization of the sum of the weighted \( p_{rt} \) terms. Constraint (18) means that we consider the penalty of ignoring the off-shifts requested by residents. Constraint (19) means that the number of shifts per month is equal for all residents. Constraint (20) means that the upper bound on the number of weekend and holiday shifts per month is equal for all residents. Constraint (21) means that the lower bound on the number of weekend and holiday shifts per month is equal for all residents. In constraints (19) – (21), we doubled the weight applied to a shift if the resident is assigned to the ICU on a holiday. Constraint (22) means that the number of shifts is balanced across the days of the week. Constraint (23) means that four residents are assigned to the critical care center on each day.

Constraint (24) means that the number of first-year residents assigned to the critical care center per day exceeds the lower bound. Constraint (25) means that the number of first-year residents assigned to the critical care center per day is below the upper bound. Constraint (26) means that the number of second-year residents assigned to the critical care center per day exceeds the lower bound. Constraint (27) means that the number of second-year residents assigned to the critical care center per day is below the upper bound. Constraint (28) means that one resident is assigned to the ICU per day shift. Constraint (29) means that residents are not assigned weekend or holiday day shifts in the ICU. Constraint (30) means that residents who rotate to anesthesiology are assigned no more than two consecutive shifts. Constraint (31) means that no resident is assigned a shift if they have plans such as a normal shift, a luncheon, or a meeting. Constraint (32) means that the lower bound on the number of shifts per month is equal for all residents. Constraint (33) means that there should be no more than one Saturday day shift per month. Constraint (34) means that there should be no more than one Saturday night shift per month. Constraint (35) means that there should be no more than two day shifts per month. Constraint (36) means that there should be no more than one ICU shift per month. Constraint (37) means that a resident should have more than three off-shifts. Constraint (38) means that there should be no more than one weekend or holiday shift per week. Constraint (39) is a non-negative constraint. Constraint (40) is a binary constraint.

We can classify these constraints into three groups. Constraint (38) arises from the Labor Standards Act. Constraints (24), (29), (30), (31), (32), and (33) arise from the rules of Aichi Medical University Hospital. Constraints (19), (20), (21), (22), (23), (25), (26), (27), (28), (34), (35), (36), (37), and (39) reflect the wishes of the staff schedulers. We separate out shifts in the ICU and the critical care center, because those in the ICU place a particularly heavy burden on the resident.
3.3 Determining the Priority of Constraints using AHP

The AHP was used to determine the priority of the constraints represented by $\alpha$, $\beta$, $\gamma$, and $\delta$ in the objective function (17) of the problem. Following the standard AHP method, evaluation criteria and alternatives were established to clarify the overall goal, which was defined in this study as that of “deciding the priority of constraints”. The evaluation criteria were determined by interviewing two individuals who are responsible for creating the schedule at Aichi Medical University Hospital, Dr. A and Mr. T, and residents of the postgraduate clinical training center. Four evaluation criteria were used: Job, Health, Salary, and Personal. “Job” refers to the difficulty of the job content across three consecutive days, “Health” refers to a lack of sleep due to overwork, “Salary” refers to payments for overtime and holiday work, and “Personal” refers to time taken off for personal travel or other leisure pursuits. Four alternative items were as follows.

Alternative 1 (A1): Consideration of the off-shifts requested by residents (i.e., constraint (18)).

Alternative 2 (A2): Equalization of the number of shifts across residents (i.e., constraint (19)).

Alternative 3 (A3): Equalization of the number of weekend and holiday shifts across residents (i.e., constraints (20) and (21)).

Alternative 4 (A4): Balancing the number of weekday shifts (i.e., constraint (22)).

Fig. 4 shows the AHP hierarchy diagram. Decisions were made by calculating the evaluation values from the lower level of Fig. 4 to the upper level. We obtained the values of the evaluation criteria and the alternatives by calculating the eigenvectors of pairwise comparison matrices.

A pairwise comparison questionnaire was administered to 27 second-year residents (13 women, 14 men). To compare two alternatives, the ratios of alternatives were estimated using a 1-18 fundamental scale of numbers. The consistency of the pairwise comparison questionnaire was judged using a consistency index that became smaller than 0.1 in all pairwise comparison matrices. This ensured that the paired comparison questionnaires used in the study were consistent. The overall evaluation values of each alternative were (A1) 0.460281, (A2) 0.21571, (A3) 0.181196, and (A4) 0.142152. Figure 5 shows a radar chart of the average overall evaluation values. It can be seen that the order of priorities was (A1), (A2), (A3), and (A4), with (A1) representing approximately 50% of all the alternatives. Before administering the questionnaire, we interviewed Dr. A and Mr. T to determine their priorities, which were found to have the order (A3), (A1), (A2), and (A4). This suggested that Dr. A and Mr. T set priorities on the alternatives that were different from those of the residents. We determined the priority of constraints to be $\alpha = 0.460281$, $\beta = 0.21571$, $\gamma = 0.181196$, and $\delta = 0.142152$.

3.4 Evaluation of the NSSR

We next discuss night-shift scheduling for February 2015. Running the scheduling optimization took approximately 5 s of CPU time. We compared three schedules:

(a) manual schedule,

(b) NSSR schedule without AHP constraint prioritization, and

Fig. 5 Average overall evaluation values.
3.3 Determining the Priority of Constraints and the alternatives by calculating the eigenvectors

Alternative 3 (A3): Mr. T, and residents of the postgraduate clinical unit at Aichi Medical University Hospital, Dr. A and criteria were determined by interviewing two individuals. The goal, which was defined in this study as that of "de-

The AHP was used to determine the priority of using AHP (i.e., constraints (20) and (21)).

Restraint (19)).

ber of shifts across residents (i.e., constran-

straints (18)).

3.4 Evaluation of the NSSR

0.142152.

= 0.460281.

determined the priority of constraints to be

and Mr. T had set priorities on the alternatives.

priorities, which were found to have the order (A3),

we interviewed Dr. A and Mr. T to determine their priorities, which were found to have the order (A3), (A2), (A1), and Mr. T set priorities on the alternatives. Before administering the questionnaire,

(A1) representing approximately 50% of all the al-

ternatives. Before administering the questionnaire,

values of each alternative were (A1) 0.460281, (A2)

of priorities was (A1), (A2), (A3), and (A4), with

study were consistent. The overall evaluation val-

sistency index that became smaller than 0.1 in all

comparison questionnaire was judged using a con-

men). To compare two alternatives, the ratios of al-


case.

3.5 Feedback from the Administration of Aichi Medical University Hospital

The NSSR has been in trial use since November 2015. Feedback has been obtained from Dr. A and Mr. T, and the formulations re-adjusted accordingly. This feedback can be summarized as follows.

(2018) The total time, including data input and calculation time, required for scheduling using the NSSR was approximately 15 min. The schedules generated by the NSSR could not be accepted because some schedules assigned residents to

the same work duties at the same shift time every week. For instance, a resident was assigned critical care day shifts on two consecutive Saturdays. Constraints (32) and (33) were added to the model to adjust the number of Saturday night shifts. This produced schedules with different night shifts, solving the problem.

(December 2015) Two schedules were generated in December 2015. The total time required was again approximately 15 min. The night-shift schedule produced by the NSSR reached the level required for actual operation. However, an experienced scheduler found ways of im-
Table 3  Number of day shifts in the week per resident in February 2015 (column: day of the week; row: name of residents (masked)): (a) manual schedule.

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>a</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>2</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>b</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>2</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>c</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>d</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>2</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>e</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>f</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>g</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>h</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>i</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>j</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>2</td>
<td>1</td>
</tr>
<tr>
<td>k</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>l</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>2</td>
<td>0</td>
</tr>
<tr>
<td>m</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>n</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>o</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>p</td>
<td>0</td>
<td>0</td>
<td>2</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>q</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>2</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>r</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>s</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>t</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>u</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>v</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>w</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>x</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>y</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>2</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>z</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>2</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>a'</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>b'</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>c'</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>2</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>d'</td>
<td>3</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>e'</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>f'</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>g'</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

proving it. The NSSR was still assigning an unequal number of work duties to some of the residents. For example, one resident was assigned to the ICU three times in one month, whereas another was assigned only once. As a first response, constraints (19)–(21) were added, to treat work in the critical care center and the ICU separately. If a resident was assigned to the critical care center, that resident was assigned a single shift. If a resident was assigned to the ICU, that resident was assigned two shifts. The resulting schedules reduced the amount of ICU overlap for a resident and helped to solve the problem. Feedback received from Dr. A suggested that the schedule was usable, while Mr. T also reported being able to make an almost complete schedule, which would have been usable in practice after some manual readjustment. However, this was not ready in time for the deadline for allocating the schedule to

Table 4  Number of day shifts in the week per resident in February 2015 (column: day of the week; row: name of residents (masked)): (b) NSSR schedule without constraint prioritization.

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>a</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>b</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>c</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>d</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>e</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>f</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>g</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>h</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>i</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>j</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>k</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>l</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>m</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>n</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>o</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>p</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>q</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>r</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>s</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>t</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>u</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>v</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>w</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>x</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>y</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>z</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>a'</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>b'</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>c'</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>d'</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>e'</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>f'</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>g'</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>
the residents.

(2016) In January 2016, Mr. T took a night-shift schedule produced by the NSSR and manually readjusted it, switching some shifts based on the operations undertaken by the department and the resident allocated to it. This took about two hours. Based on Mr. T’s experience, we added the “Check for contradictions” button, allowing the user to check for contradictions between the set data and the constraints of the model. (February 2016) For the February 2016 schedule, Mr. T again made some manual readjustments. It was confirmed that the schedule produced by the NSSR was superior to the one created manually, as it was able to satisfy all the off-shift requests of the residents, which the NSSR sets automatically. Overall, the time required for scheduling was reduced. At Mr. T’s request, we altered the layout of the NSSR, and automated some of the calculations; for example the number of weekday and weekend shifts allocated to each resident. (March 2016) The March 2016 schedule required manual readjustment that took one and a half hours.

4 CONCLUSIONS

In this study, we introduced two systems, the RSSR and NSSR, to address the RSP at Aichi Medical University Hospital. The RSSR proposed allows schedules to be created that takes the rotation to departments requested by residents and the number of residents assigned to each department into account. The amount of manual scheduling required has been significantly reduced.

The night-shift schedules obtained from the NSSR equalize the weekend and holiday shifts allocated to each resident and balance of the number of weekday shifts. In addition, we applied AHP to determine the priorities of penalty considered by residents and introduced constraints that reflected these. The schedules provided by the NSSR were also found to reduce the time spent on manual scheduling. Before introduction of the system, it would take a manual scheduler 24 hours to create the monthly schedule. By applying the NSSR, this has been reduced to approximately two hours.

The RSSR is currently being trialed by staff schedulers at the hospital. Feedback from this trial will be used to further increase the effectiveness of the system. The NSSR has been in use at Aichi Medical University Hospital since January 2016, and is being used to create the rotation schedules for new residents in April of each year. Continuous improvement in the NSSR will be made based on the experience gained from these trials.
ACKNOWLEDGMENT

This research was supported by JSPS KAKENHI Grant Numbers JP 16H07226, 16K01268.

REFERENCES