A Memetic Algorithm with Recovery Scheme for Nurse Preference Scheduling

Chun-Cheng Lin*, Jia-Rong Kang, Tzu-Hsuan Hsu
Department of Industrial Engineering and Management
National Chiao Tung University,
Hsinchu 300, Taiwan

Abstract

The previous works on designing evolutionary algorithm for nurse preference scheduling problems hardly realized that too many strict constraints on nurse preferences adopted at the same time lead to infeasible solutions frequently during the computational process. Therefore, to efficiently handle infeasible solutions, this paper proposes a memetic algorithm that incorporates the genetic algorithm, the recovery scheme, and the local search designed specifically for our nurse preference scheduling problem, which aims to maximize the total satisfaction of the nursing staff with their preference rights and interests, including each nursing staff member's preferences and preference priority ordering for work shifts and days-off, under the hard constraints of manpower demands and the number of days-off. Our experimental results for the case in a real hospital show that our nurse schedule not only fairly accomplishes the assignment of most nursing staff members to their preferred work shifts and days-off, but also satisfies all constraints.

Keywords: nurse scheduling; genetic algorithm; preference rank

---

* Corresponding author: cclin321@nctu.edu.tw

1. Introduction

The nurse schedule that takes into account the nursing staff's preferences on their own working rights and interests is helpful to promote their working efficiency and the manpower retention, and hence, has received a lot of attention recently [5], [19]. In general, such a nurse schedule (a.k.a., a nurse preference schedule) is to determine the combinations of the preferred work shifts and days-off of each nursing staff member within the planning schedule period (a week or a month generally). Note that work shift rotations are generally classified into 2-shift rotation (day and night shifts) or 3-shift rotation (day, night, and evening shifts) [4]. The arranged combinations have to satisfy a variety of constraints on laws and contracts for nurse scheduling, e.g., the skill qualification and the working time, the minimum number of staff members required in each work shift, and so on. Note that the constraint that must be satisfied is called a hard constraint, while the one that may not be satisfied but has to be punished is called a soft constraint [7], [8].

The nurse scheduling with preferences is a challenge for the schedule planner, because the one without preferences was originally a complex combinatorial optimization problem, and it is difficult and time-consuming to plan the nurse schedule manually. To efficiently plan nurse (preference) schedules with good quality, a lot of methods have been proposed recently, e.g., fuzzy approach [24], integer programming [6], [13], [18], genetic algorithm (GA) [1], [2], [26], and particle swarm optimization (PSO) [3], [14]. Most of the previous works on designing evolutionary algorithms for nurse scheduling problems applied the following strategy: first, to determine a large pool of the possible combination patterns of work shifts and days-off, and then to develop a variety of evolutionary algorithms to find the optimal pattern in this pool as the final solution, while the objective is to minimize the total penalty, including the non-preferred work shifts and days-off and those soft constraints that can be violated [15], and to randomly choose one of the other patterns if any hard constraint is violated. However, when the number of hard constraints for nurse scheduling is increased, or when the pool of the possible combination patterns of work shifts and days-off cannot be adopted effectively, the probability of generating infeasible solutions would become very high, especially, for those new solutions generated by the evolutionary algorithm, so that the times of randomly choosing one of the other patterns is drastically increased and further the algorithm is computationally inefficient.

In light of the above, it is necessary to design an efficient recovery scheme to repair those infeasible solutions, rather than discarding them. However, to the best of our understanding, there were few previous works that investigated the recovery of infeasible solutions. Therefore, this paper proposes a memetic algorithm (MA) that integrates the genetic algorithm (GA), the recovery scheme, and the local search designed specifically for our nurse preference scheduling problem. The objective of our problem is to maximize the total satisfaction of the nursing staff, in which the preferred work shifts and days-off of each nursing staff member are considered and have a priority ordering to be satisfied. The constraints of our problem focus on some common and necessary hard constraints including the manpower demand of each work shift in each day, and the lower bounds of the resting time between two shifts and the number of days-off of the each nursing staff member. If some constraints are violated in our nurse schedule, a two-stage method (first for the work shifts, and then for the days-off) is adopted to recover the solutions flexibly, to ensure the feasibility of the schedule. Finally, this paper gives the experimental results for a numerical example in a real hospital, and solves it with the conventional GA and our proposed MA. As compared with the optimal solutions, the experimental results show that our proposed MA not only performs better than the conventional GA, but also has a small error of the optimal solution (about 4%) and runs very efficiently (within about 10 minutes).

The main contributions of this paper are stated as follows:

- A novel satisfaction function of the nursing staff is proposed in a precise mathematical form. This function can not only reflect the preferred work shifts and days-off of each nursing staff member, but also take into account the preference priority ordering of each nursing staff member from the historical scheduling data to balance the total number of the satisfied work shifts and days-off for the nurse schedules in the recent several periods.
- In our proposed MA, a recovery scheme is proposed to fully utilize the redundant manpower to recover the infeasible solutions, to improve the computational efficiency of the algorithm. Note that, as recovering the infeasible solutions in evolutionary algorithms is significant for improving the effectiveness of the solution-searching process, a variety of recovery schemes in GA have been developed in other fields (e.g., highway alignment problem [16]) in the previous literature.

The rest of this paper is organized as follows. Section 2 reviews the previous works related to this paper. Section 3 describes the nurse preference scheduling problem and establishes its mathematical model. Section 4 proposed our MA for this problem. Section 5 gives the experimental setting and experimental results. A conclusion is given in Section 6.

2. Literature Review

This section reviews a variety of methods for nurse scheduling problems in the literature, including mathematical programming methods, heuristics, and metaheuristics. On mathematical programming methods, Felici and Gentile [12] established an integer programming model that maximizes the total satisfaction of the nursing staff. Bard and Purnomo [6] established a mathematical programming model based on the set covering problem that minimizes the penalty of violating nursing staff members' preferences, and adopted the column generation method to solve the model. Topaloglu and Selim [25] proposed a fuzzy multi-objective integer programming model that considers a variety of uncertain factors on nurse schedules and the nursing staff's preferences. M'Hallah and Alkhabbaz [20] proposed a mixed-integer programming model for the nurse scheduling problem in a health care unit, and adopted a case study to show their outperformance over the nurse schedules planned manually. Smet et al. [23] proposed a generic mathematical model for nurse schedule problem, which is based on the common elements considered in most previous works and the hospital constraints that are usually neglected. Maenhout and Vanhoucke [18] built a linear integer programming model to plan the nurse schedule in which nurses are allocated to different wards under the constraints of nurse staffing policies, shift scheduling policies of each ward, and the nurses' characteristics. Fan et al. [13] planed a practical nurse scheduling problem with a binary integer programming model, which aims to maximize the satisfaction of all nurses and considers seven shifts with either 8.5 or 12.5h of time lengths in the workday and some hard and soft constraints. Moreover, the authors also implemented two types of preferences for nurses. Wright and Mahar [28] solved a centralizing scheduling problem with nurses from multiple units (e.g., surgical, medical, intermediate cardiac units) in a hospital by an integer programming method, which considers two objectives: minimization of the scheduling cost (including regular time wages and overtime wages) and the minimization of the scheduling desirability.

In recent decades, it has been popular to solve the nurse scheduling problem by heuristics or evolutionary algorithms. Maenhout and Vanhoucke [19] solved the nurse scheduling problem by an electromagnetic algorithm incorporated with three local search methods for variable neighborhood. Bai et al. [5] proposed a hybrid evolutionary algorithm with the local search based on simulated annealing hyper-heuristic (SAHH) for the nurse scheduling problem, and the experimental results showed that their approach performs better in 52 instances. Hadwan et al. [15] proposed a harmony search algorithm for the nurse scheduling problem for a hospital in Malaysia. Todorovic and Petrovic [24] developed a bee colony optimization approach for the nurse scheduling problem that can eliminate some bad solutions in the solution neighborhood. Constantino et al. [11] developed a new deterministic heuristic algorithm to tackle a nurse scheduling problem, in which two phases (constructive phase and improvement phase) are used sequentially. Their experiments in almost 250,000 instances from the NSPLib dataset revealed that the results produced by the algorithm are better than previous works. Gao and Lin [14] constructed a mathematical model for the nurse scheduling problem, which aims to maximize the degree of nurses' working happiness, and considers the scheduling constraints as well as the hospital regulations at the same time. Then, they applied the conventional PSO to solve the problem to replace the time-consuming manual scheduling. Maenhout and Vanhoucke [17] proposed an algorithm based on the vertebrate immune system to cope with the announced nurse schedule with violated constraints, and compared the algorithm with GAs and a multi-start heuristic from the existing literature. The experimental results indicated that the effectivity of the proposed algorithm is promising for handling the nurse rescheduling problem. Wright and Vanhoucke [29] applied an evolutionary algorithm and a branch-and price approach from the existing literature to solve a nurse rescheduling problem in which external nurses from other departments can be called up to cope with the violated constraints in the schedule.

Based on the above, this paper first develops a mathematical programming model to find the exact optimal solutions for smaller-scale nurse scheduling problems, and then proposes an evolutionary algorithm to efficiently find the nearly optimal solutions for larger-scale nurse scheduling problems.

3. The Nurse Preference Scheduling Problem

This section introduces the nurse scheduling problem of a medical department in a real large-scale hospital, including the description of the nursing staff and work shifts, and the objective and constraints of the concerned nurse schedule. In addition, the problem is modelled as a mathematical programming model.

3.1 Problem description

The concerned department contains 20 nursing staff members in total, each of who is a full-time employee with the same skill qualification, i.e., the floating staff or the call staff are not considered. Those nursing staff members are assigned to a nurse schedule with a period of 28 days (4 weeks), in which each day has three work shifts including day shift (8:00 am – 4:00 pm), evening shift (4:00 pm – 0:00 am), and night shift

(0:00 am – 8:00 am), and there is a day-off schedule. In our nurse schedule, the work shift type of each nursing staff member in each day has flexibility, i.e., day, evening, and night shifts can be mixed to be assigned. However, in order to make each nursing staff member keep regular hours, a soft constraint of our problem is to keep the work shift type of each day as the same as possible.

The objective of our nurse preference scheduling problem is to maximize the total satisfaction of all the nursing staff members with this schedule. To ensure that the nurse schedule is feasible and most nursing staff members are satisfied, when assigning those nursing staff members to work shifts and days-off, we have to consider the following two points:

- The hard constraints for work shifts must be satisfied fully;
- According to each nursing staff member's historical schedule data and preference priority ordering, the nursing staff members are fairly assigned to their most preferred work shifts and days-off as much as possible.

The details of the constraints for work shifts and work preferences are given as follows.

3.1.1 Constraints for work shifts

The nurse schedule generally has a lot of constraints, which are concerned about laws and contracts for nurse scheduling, and hence must be satisfied fully. In this paper, the constraints that must be satisfied in the nurse schedule are called hard constraints as follows:

- Each nursing staff member is arranged to at least one work shift or one day-off in each day.
- The lower bound of the manpower demand for each work shift in each day must be satisfied. In this paper, the lower bound for day, evening, and night shifts are set to 5, 4, and 3, respectively.
- The resting time of each nursing staff member between any two shifts is at least 12 hours, to ensure that each nursing staff member has sufficient time to rest.
- The total number of days-off of each nursing staff member within the schedule period must be the same.
- Each nursing staff member must be arranged with exact two days-off in each week.

3.1.2 Work preferences and the preference priority ordering of the nursing staff

To ensure that the nursing staff have preference rights and interests in nurse schedules, before planning the nurse schedule, each nursing staff member can base her/his own preferences to subscribe the preferences of work shifts (assuming three preference ranks for work shifts: good, normal, and bad) and the two preferred days-off in each week.

In addition, to guarantee fairness of nurse schedules, the preferred work shifts and days-off of each nursing staff member have a preference priority ordering to be satisfied. The preference priority ordering is determined by the so-called *preference weight* in this paper. We assume that the nursing staff member with a larger preference weight not only is arranged to more preferred work shifts, but also has a higher probability to be arranged to more preferred days-off. The preference weights for work shifts (W_i^S) and days-off (W_i^H) of each nursing staff member i are respectively calculated as follows:

$$W_i^S = \left(\frac{T_1 \cdot \lambda_{i1}^S + T_2 \cdot \lambda_{i2}^S + T_3 \cdot \lambda_{i3}^S}{(K - \beta) / \beta} \right)^C \tag{1}$$

$$W_i^H = (L \cdot \lambda_i^H)^C \tag{2}$$

where

- $T_1 = 0$, $T_2 = 1$, and $T_3 = 2$ are the scores of work shifts for good, normal, and bad preference ranks, respectively;
- $L = 2$ is the score of the preferred day-off;
- λ_{i1}^{S}, λ_{i2}^{S}, and λ_{i3}^{S} are the times of the satisfied work shifts of staff member i for good, normal, and bad preference ranks, respectively, in the nurse schedule during the previous period;
- λ_{i}^{H} is the times of the satisfied days-off of staff member i in the nurse schedule during the previous period;
- K is the total number of the planning days in a nurse schedule;
- β is the total number of days-off of each nursing staff member in a planning period;
- $C = 2$ is the index number used to raise the power of the weight.

Consider the extreme cases where all the preference ranks of a nurse are set to be 'all good', 'all bad', and 'all normal. If all the preference ranks of a nurse for work shifts are set to be 'all good' at the current period, then the times of the assignments to her/his preferred work shifts at the current period is the greatest, so that the preference weight at the next period will be the lowest by Equation (1), i.e., this nurse will be almost not able to be assigned to her/his preferred work shifts at the next period. The case for ‘all bad’ is similar. As for the case of ‘all normal’, the nurse preference scheduling problem is degenerated to a common nurse scheduling problem. As a consequence, the numbers of the three preference ranks of a nurse are restricted in our model.

3.2 Mathematical programming model

This subsection establishes the mathematical programming model for our concerned problem, including the problem objective of maximizing the total satisfaction of the nursing staff and all constraints.

Indices:

- i is the index for a nursing staff member.
- j is the index for a work shift.
- k is the index for a day.

Parameters:

- I denotes the set of the nursing staff.
- J is the set of work shifts, and J = {1 (day shift), 2 (evening shift), 3 (night shift)}.
- K is the total number of days in a planning period.

- α is the preference coefficient, and $\alpha > 1$.
- β is the total number of days-off of each nursing staff member within the planning period.
- W_i^S is the preference weight of work shifts of nursing staff member *i*.
- W_i^H is the preference weight of days-off of nursing staff member *i*.
- $P_{i,j}^S$ is the preference satisfaction of nursing staff member *i* for work shift *j*.
- $P_{i,k}^H$ is the preference satisfaction of nursing staff member *i* for day-off *k*.

Decision variables:

- $s_{i,j,k}$ decides whether staff member *i* is assigned to work for shift *j* on day *k*, and $s_{i,j,k} \in \{0 \text{ (no assigned)}, 1 \text{ (assigned)}\}$.
- $h_{i,j,k}$ decides whether member *i* is assigned to have a day-off in shift *j* on day *k*, and $h_{i,j,k} \in \{0 \text{ (no assigned)}, 1 \text{ (assigned)}\}$.

The objective of our mathematical programming model:

$$\text{Maximize } Z = \frac{\sum_{i \in I} \sum_{j \in J} \sum_{k \in K} \left[P_{i,j}^S \cdot (s_{i,j,k} - h_{i,j,k}) \right] + \sum_{i \in I} \sum_{j \in J} \sum_{k \in K} (P_{i,k}^H \cdot h_{i,j,k})}{\left\{ \sum_{i \in I} \left(\frac{\alpha W_i^S \cdot (K - \beta)}{(K - \beta) / \beta} + \alpha W_i^S \cdot \beta \right) \right\}} \qquad (3)$$

where

$$P_{i,j}^{S} = \begin{cases} \alpha \cdot (\dfrac{W_i^S}{(K-\beta)/\beta}), & \text{if the perference rank is good;} \\ (\dfrac{W_i^S}{(K-\beta)/\beta}), & \text{if the perference rank is normal;} \\ 0, & \text{if the perference rank is bad;.} \end{cases}$$

$$P_{i,k}^{H} = \begin{cases} \alpha \cdot W_i^H, & \text{if staff member } i \text{ prefers to take day } k \text{ off;} \\ 0, & \text{otherwise.} \end{cases}$$

The constraints of our mathematical programming model:

$$\sum_{j \in J} s_{i,j,k} = 1 \ , \ \forall i \in I \tag{4}$$

$$\sum_{i \in I} \left((s_{i,j,k} - h_{i,j,k} \right) \geq D_{j,k}, \ \forall j \in J, k \in K \tag{5}$$

$$s_{i,2,k} + s_{i,1,k+1} <= 1, \quad \forall i \in I, \quad k \in K \tag{6}$$

$$s_{i,3,k} + s_{i,1,k+1} <= 1, \ \forall i \in I, \ k \in K \tag{7}$$

$$s_{i,3,k} + s_{i,2,k+1} <= 1, \ \forall i \in I, \ k \in K \tag{8}$$

$$\sum_{j \in J} \sum_{k \in K} h_{i,j,k} = \beta \ , \ \forall i \in I \tag{9}$$

$$s_{i,j,k} - h_{i,j,k} \geq 0 \ , \forall i \in I, j \in J, k \in K \tag{10}$$

$$\sum_{j \in J} \sum_{k=1}^{7} h_{i,j,k} = 2 \ , \forall i \in I \tag{11}$$

$$\sum_{j \in J} \sum_{k=8}^{14} h_{i,j,k} = 2 \ , \forall i \in I \tag{12}$$

$$\sum_{j \in J} \sum_{k=15}^{21} h_{i,j,k} = 2 \ , \forall i \in I \tag{13}$$

$$\sum_{j\in J}\sum_{k=22}^{K} h_{i,j,k}=2 \ ,\forall i\in I \tag{14}$$

$$s_{i,j,k} \in \{0,1\} \ , \forall i\in I, j\in J, k\in K \tag{13}$$

$$h_{i,j,k} \in \{0,1\} \ , \forall i\in I, j\in J, k\in K \tag{14}$$

Objective (3) is to maximize the total satisfaction of the nursing staff for the nurse schedule. The value of the objective falls between 0 and 1. If the value approaches 1, it means that the total satisfaction is higher. Note that the design of this objective tends to prioritize the nursing staff members that are not satisfied with the nurse schedule of the previous planning period to be assigned to their preferred work shifts and days-off, so as to raise the satisfaction and fairness of the nursing staff for the nurse schedule.

On our constraints, Constraint (4) enforces each nursing staff member to be assigned to a work shift in each day. Constraint (5) enforces that the lower bound of the manpower demand of each work shift of each day must be satisfied. Constraints (6) – (8) ensures that each nursing staff member must rest for no less than 12 hours between any two work shifts. Constraint (9) enforces that the total number of days-off of each nursing staff member within a planning period is the same. Constraint (10) enforces that each nursing staff member must be assigned to a work shift or a day-off in each day. Constraints (11) – (14) enforces that each nursing staff member must be assigned to take 2 days off in a week. Constraints (15) and (16) enforce that the decision variables for work shifts and days-off are binary.

4. Our Memetic Algorithm

The nurse scheduling problem has been shown to be NP-hard [27]. Hence, the schedule planner cannot manually enumerate all possible cases, nor adopt a mathematical programming method to efficiently solve large-scale nurse scheduling problems.

Recently, one of the most successfully proposed metaheuristic algorithms for a variety of applications is the memetic algorithm (MA), which generally integrates local searches into evolutionary algorithms (e.g., PSO, GA, and so on), i.e., the MA incorporates both global search and local search schemes to increase the computational efficiency of the algorithm [9], [10], [21], [22]. Hence, this paper proposes a MA with some delicate designs to solve our concerned nurse scheduling problem. In what follows, we first describe our MA, then explain its features and flowchart, and finally give the details of the MA.

As shown in Figure 1, our proposed MA includes three main components: the GA, the recovery scheme, and the local search specifically design to actively repair the infeasible solutions of our concerned problem, to ensure that the generated solutions fully satisfy all the hard constraints of nurse schedules. Hence, our MA is also called an MA with a recovery scheme, and the flowchart of our MA is given in Figure 1, which are explained as follows.

Our MA is operated with a population of chromosomes. First, the population of chromosomes are initialized randomly, and the fitness value of each chromosome is evaluated. Then, the process of a genetic algorithm (GA) is executed. The chromosomes with better fitness values are preserved, and those with worse fitness values are eliminated by iterative improvements of multiple generations for the chromosome population, so as to find better solutions. In addition, if there is some chromosome that violates some hard constrains, then the faults in the work shifts and days-off in this chromosome will be repaired. Next, this chromosome randomly selects one of two local searches to generate a new chromosome. The above process is repeated until the termination condition is satisfied.

In the rest of this section, we give the details of the main components of our MA with a recovery scheme, including the contained GA, local searches, and termination condition. In explaining the GA, the solution representation, the solution initialization, and the fitness value are introduced.

4.1 Solution representation

In GA, a solution for our nurse preference scheduling problem is represented by a chromosome. The information of a nurse schedule contains the work shift types or days-off of each nursing staff member, and hence is encoded as an $N \times K$ matrix, in which N and K are the total number of nursing staff members and the total number of days within the planning period, respectively. Each entry in the i-th row and the j-th column of the matrix is recorded by one of D, E, N, and X, which represent that member i work for a day shift, evening shift, night shift, and take a day off in day j, respectively.

Take an example in Figure 2 to illustrate of the solution representation. In this example, there are 3 members $N_1 - N_3$ and 7 days $K_1 - K_7$ within a planning period. Consider the schedule of nursing staff member 1, which is assigned to day shifts on days $K_1 - K_4$; days-off on days $K_5 - K_6$; a night shift on K_7. Note that the number of days can be set to any number according of the practical use. We set 7 days in the example in Figure 2 for convenience of explanation and illustration

4.2 Solution Initialization

After determining the solution representation of a chromosome, this subsection introduces how to initialize a number of chromosomes as the initial parent generation.

Each chromosome in the initial generation must satisfy all hard constraints of our concerned problem, i.e., the solution represented by the chromosome must be feasible. Hence, a two-stage method (i.e., days-off are assigned first, and then the work shifts are assigned) is adopted to efficiently generate multiple feasible chromosomes as follows. First, two days-off of each week are assigned to each nursing staff member randomly, but the total number of the nursing staff members resting for each work shift cannot excess the regulated number; then, each nursing staff member is randomly assigned to a fixed work shift within the planning period, but the total number of nursing staff members that work for each work shift must be greater than the lower bound of the manpower demand of the work shift.

4.3 Fitness function

After the initial parent generation is generated, the chromosomes in this generation are processed by the GA, the recovery scheme, and the local search. The optimal chromosome is iteratively found, while the generated chromosome is kept feasibly at each iteration. This subsection introduces how to use the fitness value to evaluate the quality of a chromosome.

The objective of the problem concerned in this paper is to maximize the total satisfaction of all the nursing staff members for the nurse schedule. Hence, the objective function in (3) is used as the fitness function of our MA to evaluate the quality of each chromosome (solution). Note that the computed fitness values always fall in the range [0, 1], and the fitness value approaching one represents that the total satisfaction of the corresponding solution is better.

4.4 Genetic algorithm

This subsection introduces the GA used in our MA. GA has been one of the metaheuristics that is widely applied in a variety of scheduling problems in the recent decade. GA simulates the concepts of the natural evolution to solve problems. In the GA, a number of chromosomes with better fitness values are selected as the parent generation, and then an offspring generation of chromosomes with better fitness values are generated by processing the parent generation by selection, crossover, and mutation operators. Repeat the above process until a stable optimal solution is found.

In what follows, we introduce the crossover and mutation operations in our GA, and then explain how to repair the infeasibility of the chromosomes after the crossover and mutation.

4.4.1 Crossover

The purpose of the crossover operation is to generate some diversified chromosomes in the offspring generation. Before the crossover operation, two chromosomes are selected from the parent generation. Since the crossover of two good chromosomes may have a higher probability to generate better chromosomes, the two chromosomes to be crossovered are selected by the roulette wheel selection, in which the chromosome with larger fitness values accounts for a larger area in a roulette wheel to be selected with a higher probability.

The crossover operation includes the column crossover (for the days within the planning period) and the row crossover (for nursing staff members). Note that both the two crossovers adopt the one-point crossover. Take an example in Figure 3. Consider the column crossover of the two parent chromosomes P_1 and P_2 in Figure 3(a). Then, a column cut between 7 days is selected randomly. In the case in Figure 3(a), the column cut between days 4 and 5 is selected, and then two offspring chromosomes O_1 and O_2 are generated by swapping the 4-th and the 5-th columns for chromosomes P_1 and P_2. On the other hand, consider the row crossover of the two parent chromosomes P_1 and P_2 in Figure 3(b). Then, a row cut between 3 nursing staff members is selected. In the case in Figure 3(b), the row cut between the nursing staff members N_1 and N_2 is selected, and then two offspring chromosomes O_1 and O_2 are generated by swapping the 1st and the 2nd rows for chromosomes P_1 and P_2.

Note that to increase the diversity of the whole population of chromosomes, we have a probability to choose one of the row crossover and the column crossover to be operated.

4.4.2 Mutation

After the crossover operation, each chromosome has a probability to be mutated, to avoid the generation of local optimal solutions. Our mutation operation for the nurse preference schedule of a chromosome is explained as follows. First, a column (one of 7 days) in the chromosome that we would like to mutate is selected, and then the entries in the column is resorted. For example, as shown in Figure 4, day 4 is selected in chromosome P_1, and then the resultant chromosome O_1 is generated by resorting the entries D, D, E as E, D, D in day 4.

4.4.3 Recovery Scheme

Since there are a lot of constraints in our nurse preference scheduling problem (as described in Subsection 3.1), the crossover and mutation operations described above lead to a high probability to make the schedules represented by the resultant chromosomes be infeasible. Therefore, a recovery scheme is added to our GA to repair the infeasible chromosomes. Our recover scheme is two-stage (first for working days, and then for days-off), and is given in Algorithm 1.

Algorithm 1. Recovery scheme

```
1:  for each chromosome do
2:      if the shifts of some nursing staff member in two consecutive days is
        one of DN, ED, and NE then
3:          repair this work shift violation in two consecutive days
4:      else if the number of staff members is less than the lower bound of each
        day then
5:          repair the infeasibility with the redundant manpower
6:          cancel the nursing staff members that take days-off
7:      end if
8:      if the number of days-off in each week is not equal to 2 then
9:          repair the case where the number of days-off in each week is greater
            than 2
10:         repair the case where the number of days-off in each week is less
            than 2
11:     end if
12: end for
```

Stage 1 (Shift recovery). The shift recovery includes the recovery for work shift violation in two consecutive days (Lines 2 – 3 in Algorithm 1) and the recovery for the lower bound of the manpower demand of each day (Lines 4 – 7 in Algorithm 1). On the recovery for work shift violation in two consecutive days, since the resting time between two work shifts is at least 12 hours, the nurse schedule cannot has the following three shift patterns for two consecutive days: ED, ND, and NE. Therefore, if one of the three infeasible shift patterns occur, the second day of the pattern is taken off, e.g., ED is modified to EX.

Next, on the recovery for the lower bound of the manpower demand of each day, if the lower bound of the manpower demand of each day is violated, our recovery scheme is to make use of the redundant manpower in some work shifts to support those infeasible shifts. But if there is not redundant manpower, some of the days-off that are cancelled to support those infeasible shifts.

Stage 2 (Day-off recovery). The recovery for days-off is to ensure the satisfaction of the constraint that each nursing staff member must be arranged to take at least two days-off off in each week (Lines 8 – 11 in Algorithm 1). If the constraint is violated, the nursing staff member with more days-off in each week (greater than 2) are prioritized to be repaired, and the redundant days-off are randomly transformed into work shifts, to reduce the number of days-off. Next, the nursing staff member with more days-off in each week (less than 2) are prioritized to be repaired, and the redundant work shifts are randomly transformed into days-off, to increase the number of days-off. Note that the above recovery processes have to ensure no violation of the constraints.

4.5 Local search

After crossover and mutation, to further improve the fitness performance of each chromosome, the local search is conducted on the current feasible chromosomes found so far. The types of local search include the column local search and the row local search, and there is a probability to select one of the two local searches to be applied. The details of the two local searches are described as follows:

- Column local search: Randomly select a column (day) in the chromosome matrix, and then a new chromosome is generated by resorting the entries in the column under the constraints of our nurse scheduling problem.
- Row local search: Randomly select two rows (nursing staff members) in the chromosome matrix, and then a new chromosome is generated by swapping the two rows.

4.6 Termination condition

The termination condition is to determine the total number of iterations of our MA. Our termination condition includes whether the maximal number of iterations is achieve, and whether the fitness values have been convergent. If our MA does not yet achieve the termination condition, the operations in the main loop are repeated until the termination condition is achieved.

5. Experimental Implementation and Results

This section considers a numerical example with 20 nurses in 28-day nurse schedule from a real hospital to analyse the parameter setting of our MA, and then executes our

MA on this example. Finally, the experimental results of our MA and GA are compared with the optimal solutions for evaluating their performance.

In this numerical example, the information required in the nurse preference scheduling problem includes the information of the preferred work shifts and days-off of each nursing staff member during the current planning period in Table 1, and the preference weight of each nursing staff member for work shifts and days-off in Tables 2 – 3.

5.1 Parameter setting for MA

In the parameter setting for our MA, the number of iterations and the number of chromosomes in our MA would directly influence the computational efficiency and quality of our MA. As for the total number of iterations, too many iterations may increase the total running time and decrease the solution-searching efficiency, while too few iterations may influence the quality of the schedule. In addition, the number of chromosomes would also influence the solution quality. To efficiently obtain better fitness values in our MA, this subsection analyses the number of iterations and chromosomes.

5.1.1 Analysis on the number of iterations

To obtain a reasonable number of iterations, after evaluating the change of iteration numbers with fitness values in our MA, the convergent number of iterations is applied in the rest of our experiments. The changes on the total number of iterations and fitness values are plotted in Figure 5, in which the parameter setting of our MA are listed as follows: the number of chromosomes is 160; the crossover rate is 0.5; the mutation rate is 0.01; the number of iterations is 3000.

From Figure 5, we find that before the 800-th iteration, the fitness values arise rapidly, meaning that the solution searching during this period has remarkable improvements; after the iteration number excesses 1000, the changes of the fitness values become slowly, but still keep improved as the number of iterations grows; when the iteration number is from 2300 to 3000, the fitness value of each iteration is unchanged. Therefore, the number of iterations in our MA is set to 2300.

5.1.2 Analysis on the number of chromosomes

The goal of this experiment is to observe how the increase of the chromosome number changes the fitness value. The experiments with 10 different numbers of chromosomes (including 20, 40, 60, 80, 100, 120, 140, 160, 180, 200) are conducted with the following parameter setting: the number of chromosome is 160; the crossover rate is 0.5; the mutation rate is 0.01; each experiment is conducted with 20 times and each time has 2300 iterations. Table 4 gives the performance results using different numbers of chromosomes, including the best fitness value, the worst fitness value, the average fitness value, the standard deviation of fitness values, and the average running time.

From Table 4, we find that as the number of chromosomes grows, the average fitness values become better, but when the number of chromosomes achieves more than 80, the fitness values tend to a low improvement and the running time keeps increased. Hence, the number of chromosomes used in this paper is 80, to obtain the balance between the running time and the solution-searching efficiency.

5.2 Experimental results and performance comparison

According to the analysis in the previous subsection, the experiment in this subsection applies the following parameter setting: the number of chromosomes is 80; the crossover rate is 0.5; the mutation rate is 0.01; each experiment is executed with 20 times and each time contains 2300 iterations.

Our experimental results are given in Table 5, which shows that all the constrains for work shifts and days-off are satisfied fully, and the nursing staff members with larger preference weights are assigned to more preferred work shifts, and have a higher probability to take the preferred days off, so the nurse schedule can maintain fairness in a long term.

For evaluating the algorithm performance, all the algorithms are implemented and executed on a PC with an Intel i7-3770 CPU @ 3.40GHz with a memory of 16 GB. The performance of our proposed MA is compared with the conventional GA and the optimal solutions (generated by the mathematical programming solver of IBM ILOG CPLEX studio 12.4). In addition to the problem instance with 20 nurses, we also conduct the experiments for the problem instances with 30 and 40 nurses, which have been used commonly in the previous literatures. Our performance evaluation results are given in Table 6, which includes the best fitness value, the worst fitness value, the average fitness value, the standard deviation of fitness values, and the average running time of 20 runs.

The results in Table 6 show that our proposed MA for each problem instance always performs better than the conventional GA in terms of each evaluation measure, in which the errors of average fitness values of our MA from the optimal solutions for the problem instances with 20, 30, and 40 nurses are 2.02% (=(0.99-0.97)/0.99), 3.03% (=(0.99-0.96)/0.99) , and 4.08% (=(0.98-0.94)/0.98), respectively. Therefore, our proposed MA can plan a nurse schedule of good quality within the planning period, to decrease the schedule planner's planning time.

6. Conclusion

This paper has proposed a memetic algorithm (MA) to fairly plan a nurse schedule that maximizes the total satisfaction of the nursing staff under some constraints, which provides the schedule planners a reference and reduces their time and difficulty in planning their nurse schedules. In the nurse preference schedule, since a lot of constraints on the preferred work shifts and days-off of each nursing staff member are considered strictly, the conventional evolutionary algorithms may take much time to generate a nurse schedule without violating any scheduling constraints, and with inconsistency of the manpower assigned to each work shift in each day in the schedule. Therefore, our proposed MA has a recovery scheme designed specifically for our nurse preference scheduling problem, and can fully make use of the redundant manpower in the schedule to repair the infeasible solutions. This scheme can efficiently reduce both the time of generating feasible solutions and the redundant manpower of each work shift in each day, to raise the computational efficiency and the manpower utilization rate.

In addition, to make our proposed MA efficiently obtain nurse schedules with better quality, this paper analyses the total number of iterations and the total number of chromosomes in our proposed MA, and respectively conducts a variety of experiments using different numbers of iterations and chromosomes, to decide suitable numbers of iterations and chromosomes for a specific nurse scheduling problem. Finally, as compared with the conventional GA and the optimal solutions, it is showed that the nurse schedule generated by our proposed MA is able to satisfy the preferred work shifts and days-off of most nursing staff members.

References

[1] Aickelin, U. and K. Dowsland, “Exploiting problem structure in a genetic algorithm approach to a nurse rostering problem,” *Journal of Scheduling*, 3(3), 139–153 (2000)

[2] Aickelin, U. and K. Dowsland, “An indirect genetic algorithm for a nurse-scheduling problem,” *Computers & Operations Research*, 31(5), 761–778 (2004)

[3] Akjiratikarl, C., P. Yenradee, and P. R. Drake, “PSO-based algorithm for home care worker scheduling in the UK,” *Computers & Industrial Engineering*, 53(4), 559–583 (2007)

[4] Azaiez, M. N. and S. S. Al Sharif, “A 0-1 goal programming model for nurse scheduling,” *Computers & Operations Research*, 32(3), 491–507 (2005)

[5] Bai, R., E. K. Burke, G. Kendall, J. Li, and B. McCollum, “A hybrid evolutionary approach to the nurse rostering problem,” *IEEE Transactions on Evolutionary Computation*, 14(4), 580–590 (2010)

[6] Bard, J. F. and H. W. Purnomo, "Preference scheduling for nurses using column generation," *European Journal of Operational Research*, 164(2), 510–534 (2005)

[7] Burke, E., P. Cowling, P. D. Causmaecker, and G. V. Berghe, "A memetic approach to the nurse rostering problem," *Applied Intelligence*, 15(3), 199–214 (2001)

[8] Bilgin, B., P. Demeester, M. Misir, W. Vancroonenburg, and G. V. Berghe, "One hyper-heuristic approach to two timetabling problems in health care," *Journal of Heuristics*, 18(3), 401–434 (2012)

[9] Bao, Y., Z. Hu, and T. Xiong, "A PSO and pattern search based memetic algorithm for SVMs parameters optimization," *Neurocomputing*, 117(6), 98–106 (2013)

[10] Bouly, H., Z. Dang, and A. Moukrim, "A memetic algorithm for the team orienteering problem," *4OR*, 8(1), 49–70 (2010)

[11] Constantino, A. A., L. S. Dario, E. L. de Melo, C. F. X. de Mendonça, D. B. Rizzato, and W. Romão, "A heuristic algorithm based on multi-assignment procedures for nurse scheduling," *Annals of Operations Research*, 218(1), 165–183 (2013)

[12] Felici, G. and C. Gentile, "A polyhedral approach for the staff rostering problem," *Management Science*, 50(3), 381–393 (2004)

[13] Fan, N., S. Mujahid, J. Zhang, P. Georgiev, P. Papajorgji, I. Steponavice, B. Neugaard, and P. M. Pardalos, "Nurse scheduling problem- an integer programming model with a practical application," *Systems Analysis Tools for*

Better Health Care Delivery, Vol. 74 of Springer Optimization and Its Applications, 65–98 (2013)

[14] Gao, S. C. and C. W. Lin "Particle swarm optimization based nurses' shift scheduling," in *Proceedings of the Institute of Industrial Engineers Asian Conference 2013*, Springer Press, pp. 775–782 (2013)

[15] Hadwan, M., M. Ayob, N. R. Sabar, and R. Qu, "A harmony search algorithm for nurse rostering problems," *Information Sciences*, 233, 126–140 (2013)

[16] Kang, M. W., P. Schonfeld, and N. Yang, "Prescreening and Repairing in a Genetic Algorithm for Highway Alignment Optimization," *Computer-Aided Civil and Infrastructure Engineering*, 24(2), 109–119 (2009)

[17] Maenhout, B. and M. Vanhoucke, "An artificial immune system based approach for solving the nurse re-rostering problem," in *Proceedings of 13th European Conference on Evolutionary Computation in Combinatorial Optimization (EvoCOP 2013)*, Vol. 7832 of Lecture Notes in Computer Science, pp. 97–108 (2013)

[18] Maenhout, B. and M. Vanhoucke, "An integrated nurse staffing and scheduling analysis for longer-term nursing staff allocation problems," *Omega*, 41(2), 485–499 (2013)

[19] Maenhout, B. and M. Vanhoucke, "An electromagnetic meta-heuristic for the nurse scheduling problem," *Journal of Heuristics*, 13(4), 359–385 (2007)

[20] M'Hallah, R. and A. Alkhabbaz, "Scheduling of Nurses: A case study of a Kuwaiti health care unit," *Operations Research for Health Care*, 2(1–2), 1–19 (2013)

[21] Ma, H., D. Simon, M. Fei, X. Shu, and Z. Chen, "Hybrid biogeography-based evolutionary algorithms," *Engineering Applications of Artificial Intelligence*, 30, 213–224 (2014)

[22] Sudholt, D., "The impact of parametrization in memetic evolutionary algorithms," *Theoretical Computer Science*, 410(26), 2511–2528 (2009)

[23] Smet, P., P. De Causmaecker, B. Bilgin, and G. V. Berghe, "Nurse rostering: a complex example of personnel scheduling with perspectives," *Automated Scheduling and Planning Studies in Computational Intelligence*, 505, 129–153 (2013)

[24] Todorovic, N. and S. Petrovic, "Bee colony optimization algorithm for nurse rostering," *IEEE Transactions on Systems, Man, and Cybernetics*, 43(2), 467–473 (2013)

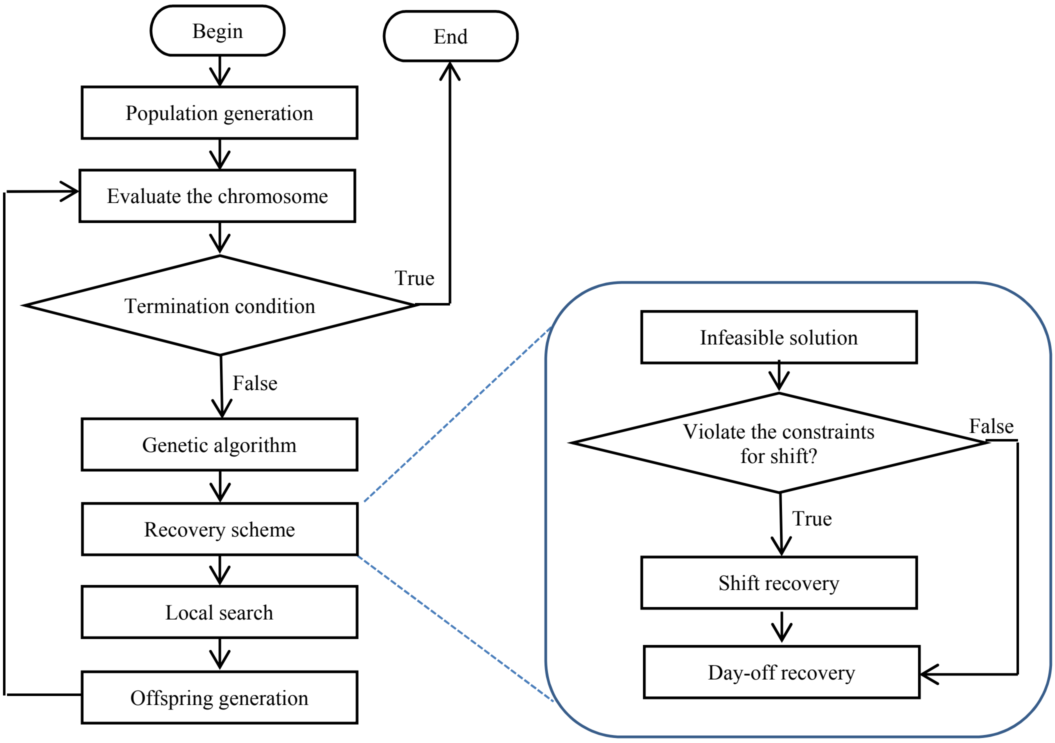
[25] Topaloglu, S. and H. Selim, "Nurse scheduling using fuzzy modeling approach," *Fuzzy Sets and Systems*, 161(11), 1543–1563 (2010)

[26] Tsai, C. C. and S. H. A. Li, "A two-stage modeling with genetic algorithms for the nurse scheduling problem," *Expert Systems with Applications*, 36(5), 9506–9512 (2009)

[27] Valouxis, C., C. Gogos, G. Goulas, P. Alefragis, and E. Housos, “A systematic two phase approach for the nurse rostering problem,” *European Journal of Operational Research*, 219(2), 425–433 (2012)

[28] Wright, P. D. and S. Mahar, “Centralized nurse scheduling to simultaneously improve schedule cost and nurse satisfaction,” *Omega*, 41(6), 1042–1052 (2013)

[29] Wright, P. D. and M. Vanhoucke, “Reconstructing nurse schedules: computational insights in the problem size parameters,” *Omega*, 41(5), 903–918 (2013)



(a) MA (b) Recovery scheme

Figure 1. The flowchart of our MA with a recovery scheme

	K_1	K_2	K_3	K_4	K_5	K_6	K_7
N_1	D	D	D	D	X	X	E
N_2	N	N	X	D	D	X	D
N_3	N	N	X	E	E	X	D

Figure 2. An example of the solution representation of a chromosome.

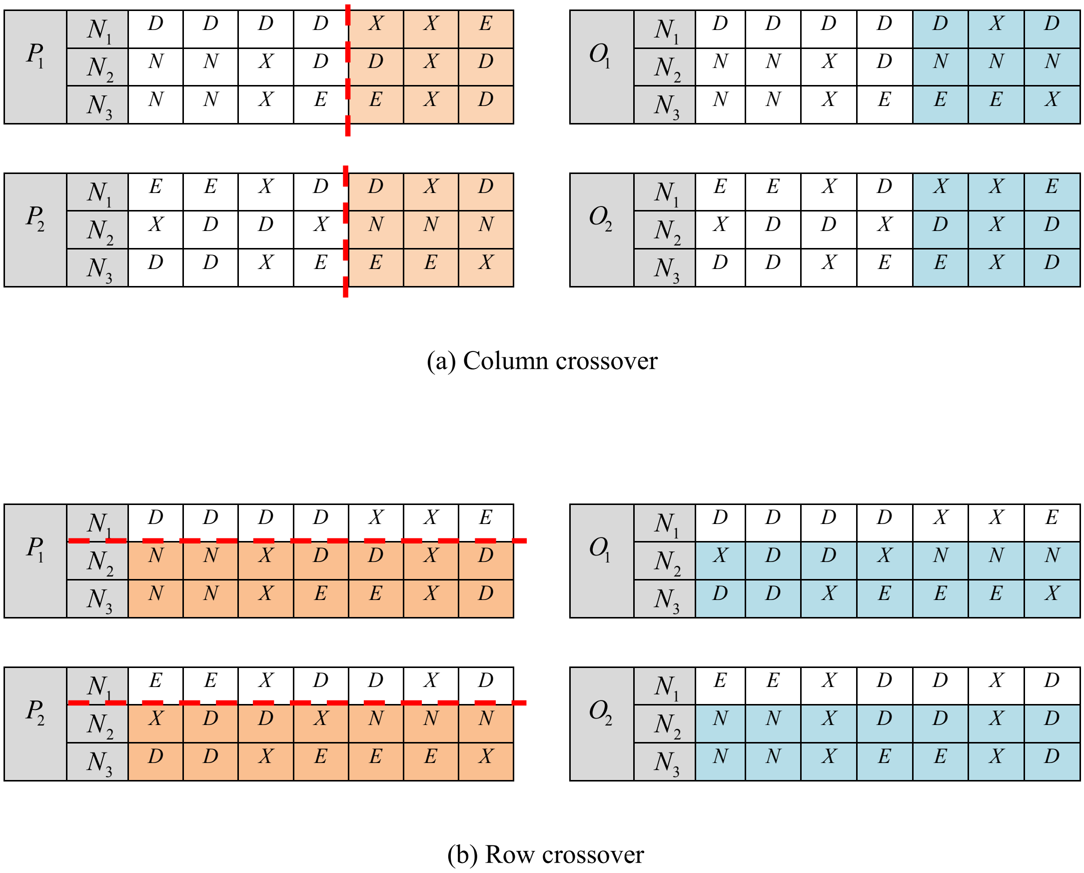


Figure 3. Illustration of two crossover operations.

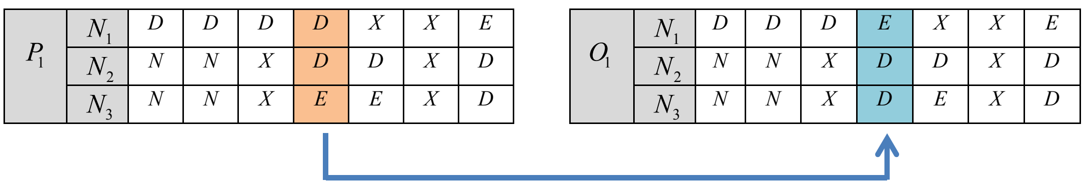


Figure 4. Illustration of our mutation operation.

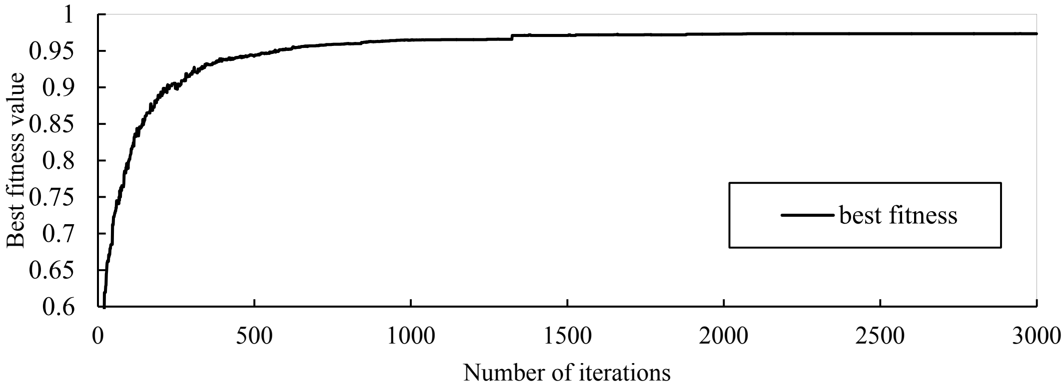


Figure 5. The plot of the best fitness value versus the number of iterations.

Table 1. The preferred ranks of work shifts and the preferred days-off of each nurse staff member.

Member No.	The preference ranks of work shifts and the preferred days-off									
	Work shift			Day-off						
	D	E	N	M	Tu	W	Th	F	Sa	Su
1	△	×	○						✓	✓
2	△	○	×				✓	✓		
3	×	△	○						✓	✓
4	○	×	△						✓	✓
5	×	△	○			✓	✓			
6	×	○	△	✓				✓		
7	×	△	○	✓	✓					
8	×	△	○	✓	✓					
9	△	×	○				✓			✓
10	○	×	△			✓			✓	
11	×	△	○	✓				✓		
12	△	×	○				✓		✓	
13	△	×	○			✓				✓
14	×	○	△		✓			✓		
15	○	△	×		✓	✓				
16	○	×	△	✓						✓
17	×	△	○			✓			✓	
18	×	△	○		✓				✓	
19	△	○	×	✓						✓
20	△	×	○			✓			✓	
The number of the most preferred	4	4	12	6	5	6	4	4	8	7
Lower bound of manpower demand	5	4	3	-	-	-	-	-	-	-

The preference rank of work shifts: ○ (Good), △ (Normal), and × (Bad); the preferred day-off: ✓

Table 2. The preference weights of work shifts of each nursing staff member.

Member No.	The times of assigning preferred work shifts in the previous period			Preference weight for work shifts
	Good	Normal	Bad	
1	18	1	1	1.44
2	16	2	2	5.76
3	12	4	4	23.04
4	16	2	2	5.76
5	18	2	0	0.64
6	20	0	0	0.00
7	8	8	4	40.96
8	18	0	2	2.56
9	12	0	8	40.96
10	0	6	14	184.96
11	8	8	4	40.96
12	16	4	0	2.56
13	0	2	18	231.04
14	4	4	12	125.44
15	16	2	2	5.76
16	17	1	2	4.00
17	8	8	4	40.96
18	9	9	2	27.04
19	2	9	9	116.64
20	0	14	6	108.16

Table 3. The preference weights of days-off of each nursing staff member.

Member No.	The times of assigning preferred days-off in the previous period		Preference weight for days-off
	Preferred	Non-preferred	
1	6	2	16
2	8	0	0
3	6	2	16
4	8	0	0
5	2	6	144
6	4	4	64
7	6	2	16
8	8	0	0
9	6	2	16
10	1	7	196
11	7	1	4
12	2	6	144
13	0	8	256
14	5	3	36
15	3	5	100
16	4	4	64
17	4	4	64
18	7	1	4
19	5	3	36
20	1	7	196

Table 4. The experimental results using different numbers of chromosomes.

Number of chromosome	Max iteration	Best	Average	Worst	StdDev	Average time (seconds)
20	2300	0.9183	0.8929	0.8477	0.0174	12.47
40	2300	0.9442	0.9306	0.9110	0.0086	26.02
60	2300	0.9780	0.9720	0.9517	0.0070	31.62
80	**2300**	**0.9833**	**0.9721**	**0.9529**	**0.0088**	**42.70**
100	2300	0.9761	0.9718	0.9619	0.0045	61.10
120	2300	0.9720	0.9685	0.9599	0.0033	64.14
140	2300	0.9855	0.9796	0.9705	0.0045	74.95
160	2300	0.9860	0.9842	0.9794	0.0023	85.31
180	2300	0.9858	0.9840	0.9779	0.0021	97.22
200	2300	0.9971	0.9875	0.9745	0.0065	119.70

Table 5. The work shift schedule.

	Planning period																											
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28
	M	Tu	W	Th	F	Sa	Su	M	Tu	W	Th	F	Sa	Su	M	Tu	W	Th	F	Sa	Su	M	Tu	W	Th	F	Sa	Su
S1	D	D	D	D	D			D	D	D	D	D			D	D	D	D	D			D	D	D	D	D		
S2	E	E		E		E	E		E		E	E	E	E	E			E	E	E	E	E			E	E	E	E
S3	E	E	E	E	E			E	E	E	E	E			E	E	E	E	E			E	E	E	E	N		
S4	D		D	X	D	D	D	D	D	D			D	D	D	D	D			D	D	D	D	D			D	D
S5	E	E			E	E	E	E	E			E	E	E	E	E			E	E	E	E	E			E	E	E
S6		D	D	D		D	D		D	D	D		D	D		D	D	D		D	D		D	D	D		D	D
S7			N	N	N	N	N			N	N	N	N	N			N	N	N	N	N			N	N	N	N	N
S8	D		D		D	D	D	D		D		D	D	D	D		D		D	D	D	D		D	X	D	D	D
S9	N	N	N		N	N		N	N	N		N	N		N	N	N		N	N		N	N	N		N	N	
S10	D	D		D	D		D	D	D		D	D		D	D	D		D	D		D	D	D		D	D		D
S11		N	N	N		N	N		N	N	N		N	N		N	N	N		N	N		N	N	N		N	N
S12	D	D	E		E	X	E	E	E	E		E		E	E	E	E		E		E	E	E	E		E		E
S13	N	N		N	N	N		N	N		N	N	N		N	N		N	N	N		N	N		N	N	N	
S14	E		E	E		E	E	E		E	E		E	E	E		E	E		E	E	E		E	E		E	E
S15	D			D	D	D	D	D			D	D	D	D	D			D	D	D	D	D			D	D	D	D
S16		D	D	D	D	D			D	D	D	D	D			D	D	D	D	D			D	D	D	D	D	
S17	N	N		N	N		N	N	N		N	N		N	N	N		N	N		N	N	N		N	N		N
S18	N		N	N	N		N	N		N	N	N		N	N		N	N	N		N	N		N	N	N		N
S19		E	E	E	E	E			E	E	E	E	E			E	E	E	E	E			E	E	E	E	E	
S20	N	N		N	N		N	N	N		N	N		N	N	N		N	N		N	N	N		N	N		N

Table 6. The performance of different algorithms for different-scale problems

Problem	Optimal solution	Method	Number of chromosomes	Max iteration	Best	Average	Worst	StdDev (%)	Average time (seconds)	Error of the optimal solution (%)
20 nurses	0.99	GA	80	2300	0.98	0.95	0.92	1.29	45.02	3.28
		MA	80	2300	0.98	0.97	0.95	0.88	53.50	1.61
30 nurses	0.99	GA	80	2300	0.91	0.89	0.86	1.59	271.58	10.10
		MA	80	2300	0.98	0.96	0.94	0.81	309.16	3.03
40 nurses	0.98	GA	80	2300	0.89	0.87	0.86	1.04	288.90	11.22
		MA	80	2300	0.95	0.94	0.93	0.57	336.42	4.08