Research Article

Nurse Scheduling with Joint Normalized Shift and Day-Off Preference Satisfaction Using a Genetic Algorithm with Immigrant Scheme

Chun-Cheng Lin, Jia-Rong Kang, Ding-Jung Chiang, and Chien-Liang Chen

1Department of Industrial Engineering and Management, National Chiao Tung University, Hsinchu 300, Taiwan
2Department of Digital Multimedia Design, Taipei Chenshih University of Science and Technology, Taipei 112, Taiwan
3Department of Computer Science and Information Engineering, Aletheia University, New Taipei City 251, Taiwan

Correspondence should be addressed to Chien-Liang Chen; clchen@au.edu.tw

Received 5 November 2014; Revised 28 January 2015; Accepted 18 February 2015

Academic Editor: Neil Y. Yen

Copyright © 2015 Chun-Cheng Lin et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

To make a fair and satisfactory nurse shift schedule, this paper proposes a novel preference satisfaction function, in which numbers of the preferred work shifts and days-off of the nursing staff are balanced, and ranks for preferences and number of the preference ranks satisfied so far are also considered. Such a preference function is capable of equivalently and fairly planning the nurse preference schedule to improve the total satisfaction. Additionally, distributed sensors can be applied to collect the information on hospital beds to provide the schedule planner to determine the lowest required amount of manpower for each work shift, to avoid the working overload of the nursing staff. To solve the nursing schedule problem, we propose a genetic algorithm (GA) with an immigrant scheme, in which utilization of the immigrant scheme is helpful in efficiently reducing amount of infeasible solutions due to practical scheduling constraints, so that the GA can efficiently find better solutions for larger-scale problems. Performance of the proposed GA with and without solution recovery scheme is evaluated by conducting a comprehensive experimental analysis on three different-size instances.

1. Introduction

In the nurse scheduling problem, work shifts and days-off are assigned to all nurses accordingly to satisfy all the demands of patients in a hospital. In the typical nurse scheduling problem, the schedule planning period is usually two weeks or one month, and the work shifts of each day are three rotations (day, evening, and night shifts) or two rotations (AM and PM shifts) [1, 2]. A lot of laws and regulations on the basic scheduling restrictions in nursing schedules exist, for example, the least required amount of the manpower at each shift and the satisfaction of the rights to take rest and days-off. To ensure making the nurse schedule feasibly satisfy all the constraints, the nurse scheduling problem is hard to be solved.

Recently, more and more attention is taken to the nurse preference scheduling problem [3], in which each nurse's preferred work shift is emphasized, so that she or he is satisfied with the planned schedule. The satisfaction of the nursing staff with their assigned work shifts and days-off can effectively raise their working morals, so that their service quality for patients can further be increased, and the health-care costs and occupational risks in hospitals can further be improved [1, 4]. Most of the nurse schedules are not only required to satisfy the basic scheduling constraints but also concerned with the preferences of the nursing staff for shifts and days-off to raise their satisfaction with the schedules. Therefore, this paper is concerned with such a type of nurse schedules.

This paper is concerned with the satisfaction of the nursing staff with the nurse schedule. Most previous related works focused on maximizing the satisfaction based on the total amount of the assignments of the preferred work shifts and days-off of the nursing staff at the current planning...
period. However, the design of such satisfaction might be unreasonable, because the numbers of the preferred work shifts and days-off are not equivalent so that the planned schedule might be biased. In light of the above, a novel satisfaction function about the preferences of the nursing staff is proposed. The satisfaction function in this paper is based upon our previous work in [5], and the main differences of this paper from our previous work are listed as follows: the numbers of the preferred work shifts and days-off are balanced; this paper allows each nurse to work for different shifts during the planning period; the preferences of days-off in each week are fixed.

In addition, to promote the satisfaction of the nursing staff with the nurse schedule, distributed sensors are applied. Distributed sensors are microsized devices, which can sense and detect the environmental objects and change and then transmit those collected data to the data collection center for further computation and analysis. These devices are used to collect the information on some real-time information, for example, the number of available beds and the health conditions of patients for each bed, to provide the schedule planner to appropriately determine the amount of the manpower of each shift in each day, so as to avoid the work overloading.

To solve the concerned problem, this paper proposes a genetic algorithm with immigrant scheme (a.k.a., GAIS). The goal of using the immigrant scheme is to effectively decrease the number of infeasible solutions due to the practical scheduling constraints, so that the GA can efficiently find better solutions for small-scale problems (i.e., with less than 20 nurses) or middle-scale problems (i.e., with 20–40 nurses). Note that the GAIS is different from the previous metaheuristics (e.g., see [6]) that commonly applied complex and time-consuming solution recovery schemes to repair the solution infeasibility [6].

Last, two standard GAs (with and without recovery scheme, resp.) and our proposed GAIS for some nurse scheduling problem instances are evaluated experimentally. Experimental results show that the GA with recovery scheme and our proposed algorithm can obtain more reasonable feasible solutions. In addition, as compared with the GA with recovery scheme and our proposed GAIS, the solutions of our GAIS perform slightly worse than the GA-II in small-scale or middle-scale problems, but the computing time is much shorter, so that the nurse schedule can be produced in a short time. Therefore, the information on whether each nurse’s preferences for work shifts and days-off are satisfied can be provided rapidly to the schedule planner to make coordination and adjustment immediately and to avoid the repetition of repairing the final schedule.

The major contributions of this paper are given as follows.

(i) This paper proposes an integer programming model that incorporates the planning of work shifts and days-off and considers the nurse scheduling regulations commonly applied to hospitals. The model ensures the achievement of the optimal solutions for small-scale problems.

(ii) The design of our preference satisfaction function is to consider the balance of the numbers of the assigned preferred work shifts and days-off, so that the function can fairly plan the preferences of work shifts and days-off to promote the total satisfaction of the schedule.

(iii) Distributed sensors are applied to realize the related real-time information, so that the schedule planner can reasonably plan the amount of the manpower of each work shift in each day to avoid each nurse’s work overloading.

(iv) This paper proposes a genetic algorithm with immigrant scheme for the concerned problem, which can generate a feasible schedule in a short time, so that nurses can obtain the scheduling results in time, and the unsatisfied nurses can immediately respond to the schedule planner to readjust and coordinate the scheduling results.

The remaining part of this paper is stated as follows. Section 2 gives an overview for nursing scheduling. Then, Section 3 gives the problem description and the mathematical model of the problem concerned in this paper. Then, Section 4 introduces how to utilize the genetic algorithm with immigrant scheme to solve the problem. Section 5 gives the experimental results, and finally, Section 6 concludes this paper.

2. Literature Review

Various nurse scheduling problems were resolved by metaheuristic algorithms or mathematical method in the past. Burke et al. [7] developed a hybrid Tabu search method for nurse scheduling problem, which can efficiently generate a schedule better than the one produced manually. Burke et al.’s work [8] is based on the Tabu search algorithm and the memetic approach to develop a hybrid memetic algorithm for the nurse scheduling problem. Maenhout and Vanhoucke et al.’s work [9] solved the nurse scheduling problem with three nursing skill levels by a novel algorithm incorporating interactive evolutionary algorithm (IEA), bacterial evolutionary algorithm (BEA), and Pareto optimality. Their experimental results showed that the incorporated algorithm can provide a nurse schedule with good quality in a short time. Ackel and Dowland [10] planned a nurse scheduling with a special day-night structure, in which nurses are contracted to work for either days or nights in one week. They proposed a genetic algorithm with a novel solution representation to drastically decrease the solution search space, and their experiments showed that the proposed algorithm performs more flexibly, efficiently, and stably than the Tabu search algorithm. Maenhout and Vanhoucke [11] proposed an improved electromagnetic algorithm that considered three different local search schemes to effectively
handle a lot of combination optimization problems. Moz and Vaz Pato [6] employed the genetic algorithm to solve a nurse rostering problem when one or more nurses cannot work in the shifts that were previously assigned to her/him or them. Bai et al. [12] proposed a hybrid evolutionary algorithm that incorporates SRGA and SAHH for the nurse scheduling problem in which a higher-grade nurse can cover the demand for a lower-grade nurse and part-time nurses. Their experimental results showed that their proposed algorithm performs better than the SAHH algorithm in 52 practical problem instances. Todorovic and Petrovic [13] developed a bee colony optimization approach that removes the worse solutions in neighborhood to solve the nurse scheduling problem with many soft constraints. Hadwan et al. [14] employed the harmony search algorithm (HSA) to plan the nurse schedule in Malaysia. Topaloglu and Selim [15] built a fuzzy multiobjective integer programming model based on the nursing staff’s preferences and a variety of uncertain factors on nurse schedules. Wright and Mahar [16] proposed an integer programming method, which considers two objectives: minimization of the scheduling cost and the minimization of the scheduling desirability, to solve a centralizing scheduling problem with nurses from multiple units in a hospital. Maenhout and Vanhoucke [17] established a linear integer programming model for nurse schedule problem, in which nurses are allocated to different wards under various constraints: nurse staffing policies, shift scheduling policies of each ward, and the nurses’ characteristics. Constantino et al. [18] applied a novel deterministic heuristic algorithm to deal with a nurse scheduling problem, in which constructive phase and improvement phase are used sequentially. Their experiments showed that their algorithm is better than previous works in almost 250,000 instances from the NSPLib dataset. Maenhout and Vanhoucke [19] solved the nurse rescheduling problem by an evolutionary algorithm and a branch-and-price approach from the existing literature for a nurse rescheduling problem, in which external nurses from other departments can be called up to cope with the violated constraints in the schedule.

3. Problem Description

This section describes the nurse scheduling problem concerned in this paper and then gives the mathematical programming model of this problem.

3.1. Assumption. The planning period of the nurse schedule is 28 days, and each day can be assigned to take a day-off or work for one of the following three shifts: day shift (AM 8:00–PM 4:00), evening shift (PM 4:00–AM 0:00), and night shift (AM 0:00–AM 8:00). Note that the day shift is the beginning shift of each day. The least required amount of the manpower of each work shift is varied and can be determined according to the information collected by distributed sensors.

The combinations of work shifts and days-off in a nurse schedule are diversified and allow each nurse to work for different work shifts during the planning period. To make all the nurses satisfied with the assigned combination in the nurse schedule, the preferences for work shifts and days-off are considered. Before planning the schedule, each nurse is asked to rank the preferences of her or his three work shifts (i.e., good, normal, and bad ranks) and the preferences of the fixed days-off in each week to be taken (e.g., to take Saturdays and Sundays off during the 28 days). In addition, the numbers of the assigned preference ranks for work shifts and days-off during the previous scheduling period, respectively, are referred to for planning the current scheduling period.

The hard constraints concerned in our problem are listed as follows.

Constraint 1. In each work shift, each nurse must work for 8 hours.

Constraint 2. Each nurse must be assigned to a work shift or a day-off in each day.

Constraint 3. For each pair of consecutive work shifts, each nurse needs to take at least 16 hours off. Hence, there are 3 infeasible pairs of consecutive work shifts as follows: the evening shift followed by a day shift, the night shift followed by a day shift, and the night shift followed by a night shift.

Constraint 4. The lowest required amount of the manpower of each work shift in each day must be satisfied.

Constraint 5. Each nurse must take 2 days-off in each week.

3.2. Mathematical Model. The model is developed based on our previous work in [5], so the parameters (i.e., \( \alpha, L, \) and C) for the model are set the same as our previous work. The main difference of this paper from our previous work is that the numbers of the preferred work shifts and days-off are balanced for the design of weights and satisfactions.

The mathematical programming model is stated in detail as follows.

Parameters

- \( n \): the total number of nurses;
- \( I \): set of nurse indices, that is, \( I = \{1, 2, \ldots, n\} \);
- \( J \): set of shift indices, that is, \( J = \{1 \text{ (day)}, 2 \text{ (evening)}, 3 \text{ (night)}\} \);
- \( K \): set of day indices, since a 28-day schedule is planned in this paper, \( K = \{1, 2, \ldots, 28\} \);
- \( i \): nurse index, that is, \( i \in I \);
- \( j \): index for a work shift, that is, \( j \in J \);
- \( k \): day index, that is, \( k \in K \);
- \( \alpha \): preference coefficient, \( \alpha = 2 \);
- \( \beta \): the total number of days-off taken by each nurse during the 28-day schedule, that is, \( \beta = 8 \);
- \( W_i^S \): the work shift weight for nurse \( i \);
- \( W_i^H \): the day-off weight for nurse \( i \);
- \( P_{i,j}^S \): the preference satisfaction of nurse \( i \) for work shift \( j \);
Mathematical Model

Maximize $Z$ subject to

$$
\sum_{i,j} (s_{i,j,k} + h_{i,k}) = 1, \quad \forall i \in I, k \in K, \quad (9)
$$

$$
\sum_{i \in I} s_{i,k} \geq D_{j,k}, \quad \forall j \in J, k \in K, \quad (10)
$$

$$
s_{i,j,k} + s_{i,k+1} + s_{i,j+1} \leq 1, \quad \forall i \in I, k \in K, \quad (11)
$$

$$
s_{i,2,k} + s_{i,3,k} + s_{i,1,k+1} \leq 1, \quad \forall i \in I, k \in K \setminus \{28\}, \quad (12)
$$

$$
s_{i,3,k} + s_{i,1,k+1} + s_{i,2,k+1} \leq 1, \quad \forall i \in I, k \in K \setminus \{28\}, \quad (13)
$$

$$
\sum_{k \in \{1,2,\ldots,7\}} h_{i,7a+k} = 2, \quad i \in I, a \in \{0,1,2,3\}, \quad (14)
$$

$$
s_{i,j,k} \in \{0,1\}, \quad \forall i \in I, j \in J, k \in K, \quad (15)
$$

$$
h_{i,k} \in \{0,1\}, \quad \forall i \in I, k \in K. \quad (16)
$$

In the proposed mathematical model above, objective (4) is to maximize the total satisfaction of the nursing staff with the planned nurse schedule. Note that the satisfaction levels fall into a range between 0 and 1, and the level approaching 1 represents that all the nurses are more satisfied with the planned nurse schedule. The satisfaction function designed here is to incorporate the preferences for both work shifts and days-off, which are calculated, respectively, by (5) and (6) (which calculate their respective weights by (7) and (8)).

The feature of the designed satisfaction function is to fairly satisfy each nurse's preferences for both work shifts and days-off. In general, if the number of the assignments of the preferred work shifts and days-off is greater, then the total satisfaction of the schedule is higher. Since the number of the planning days for work shifts is greater than the number of the planning days for days-off (i.e., there are 20 days for work shifts, while there are only 8 days for days-off), the numbers of the assigned work shifts and days-off could not be balanced, so that the preferred work shifts could be assigned with a higher priority than the preferred days-off. In this case, the number of the assignments of the preferred days-off could be decreased. As a consequence, our model employs $(K - \beta)/\beta$ to adjust the satisfaction and weight of work shifts to simultaneously satisfy the preferences of work shifts and days-off.

Constraints (9)–(16) are explained as follows. Constraint (9) enforces that each nurse must be assigned to either work for only one shift or take a day-off. Constraint (10) enforces that the least required amount of the manpower for each shift must be satisfied, in which the required amount of the manpower is computed according to the information on some real-time information (e.g., number of available beds and patients' health conditions) from distributed sensors in a hospital. Constraints (11)–(13) enforce that each nurse must take enough time off in each pair of consecutive work shifts. Constraint (14) enforces that the total number of the assignments of days-off of each nurse during the whole scheduling period must be the same. Constraint (14) enforces that each nurse in each week is assigned to take two days-off.
Constraints (15) and (16) enforce that each variable for work shifts and days-off is 0 or 1.

The conventional nurse scheduling problem is NP-hard [14, 20]. Since our nurse scheduling problem simultaneously considers the historical data of nurses and their preferences for different work shifts and days-off, the problem becomes more complex and difficult. In addition, the mathematical programming method is usually adopted to solve small-scale problems. Based on the above, a metaheuristic algorithm is developed to solve this problem.

4. Our GAIS Approach

GA [21] is a stochastic metaheuristic algorithm, which imitates the features of the natural evolution to search the global solution. In the literature, a lot of improved GAs were developed and have been shown to have good performance, for example, [22–25]. The GA with immigrant scheme (GAIS) is one of the improved GAs [26]. The main feature of GAIS is that the use of the immigrant scheme is to rapidly search better solutions in larger-scale problems.

The basic idea of GAIS is explained as follows. The basic GA evolves multiple generations of a population of multiple individuals, in which each individual represents a candidate solution for the concerned problem. In each generation (iteration) of the algorithm, each individual is improved by some evolutionary operators, for example, crossover and mutation. The GA further considers the immigrant scheme, in which some individuals in the population are replaced by immigrants; that is, those immigrants become the individuals in the next-generation population. In general, there are two types of immigrants: the random immigrants (which are generated randomly) and the elite immigrants (i.e., those best individuals in the previous-generation population).

This section proposes a GAIS for the nurse scheduling problem described in the previous section, in which the immigrant scheme is designed specifically for the concerned nurse scheduling problem. The proposed algorithm can efficiently search a better solution without any complex and time-consuming recovery scheme for middle-scale problems (i.e., with 20–40 nurses). The main steps of designing the GAIS for the concerned problem are listed as follows.

Step 1. Represent each possible solution as an individual of the population.

Step 2. Initialize the population and evaluate the fitness of each individual of the population.

Step 3. Select some parts of the current population to be handled by crossover and mutation.

Step 4. Replace the worse individuals in the current population by the immigrants with better fitness values.

Step 5. The immigrant scheme is conducted on the current population.

Step 6. Repeat the above steps until the stop criteria are achieved.

4.1. Solution Representation. In GA, each individual in the population represents a candidate solution for the concerned problem. For our concerned 28-day nurse scheduling problem, each solution is represented as an $n \times 28$ matrix (see Table 1), where $n$ denotes the number of nurses; for $i \in \{1, 2, \ldots, n\}$, $N_i$ denotes the $i$th nurse; for $j \in \{1, 2, \ldots, 28\}$, $K_j$ denotes the $j$th day of the scheduling period. Since each nurse in the schedule must be assigned to work for a shift or take a day-off in each day, each entry in the matrix is filled with one of $D$ (day shift), $E$ (evening shift), $N$ (night shift), and $X$ (taking a day-off).

For example, consider a 7-day nurse schedule. If there is a row consisting of $N, N, X, D, D, X$, and $D$, it means that the corresponding nurse is assigned to work for night shifts in days 1 and 2; take days-off for days 3 to 6; work for day shifts in the other days.

Population Initialization and Fitness Evaluation. As for the population initialization, each individual of the population is initialized randomly under the scheduling constraints.

For the fitness evaluation, a penalty function is designed to avoid the GAIS from the generation of infeasible individuals to obtain better individuals [10, 14]. Therefore, we modify objective (4) to the following fitness function:

$$
Z = \frac{\sum_{i \in I} \sum_{j \in J} \sum_{k \in K} \left( p_{i,j}^{s_i} \cdot s_{i,j,k} \right) + \sum_{i \in I} \sum_{k \in K} \left( p_{i,k}^{h_i} \cdot h_{i,k} \right)}{\sum_{i \in I} \left( \alpha W_i^S \cdot (K - \beta) / ((K - \beta) / \beta + \alpha W_i^H \cdot \beta) \right)} - M \left( \sum_{i \in I} \sum_{j \in J} \sum_{k \in K} \sum_{l \in L} V_{i,j,k,l} \right),
$$

(17)

where $l$ is the constraint index, $L$ is the set of all the scheduling constraints, $M > 1$ is a large number that penalizes the solution infeasibility, and the decision variable is

$$
V_{i,j,k,l} = \begin{cases} 
1, & \text{if nurse } i \text{ working for shift } j \text{ in day } k \\
0, & \text{otherwise.}
\end{cases}
$$

(18)

If the computed fitness value falls into the range between 0 and 1, it means that the corresponding solution is feasible. If the fitness value approaches to 1, it means that the total satisfaction of the schedule becomes better. On the other
hand, if the fitness value is negative, it means that the corresponding solution is infeasible.

4.3. Crossover and Mutation Operators. The crossover and mutation operators directly influence the solution quality of GA. The crossover operator is to swap parts of the matrices of two individuals to generate new individuals, which replace a part of the population. On the other hand, the mutation operator is to slightly modify those crossover individuals to increase the solution diversification.

In this paper, the one-point crossover operator is applied, as illustrated in Figures 1(b) and 1(c). First, two individuals (matrices) are selected from the population (see Figure 1(a)). Then, we randomly choose either one row or one column as the crossover cut, which divides each matrix into two parts, and then we swap the two parts of the two individuals to generate new individuals. The case for cutting one column is called the column crossover (Figure 1(b)), while the case for cutting one row is called the row crossover (Figure 1(c)).

After the crossover operator, we apply the mutation operator. There is a mutation rate to select those individuals to be mutated. For the selected individual to be mutated, we first choose a random row and then swap the first day-off in each week with the other days (maybe work shifts or days-off) in the week. Our mutation is illustrated in Figure 1(d), in which we select one of the two individuals in Figure 1(b) and then swap the first day-off in the first row with the three days in the first row.

4.4. Local Search. After crossover and mutation operators, local search is conducted to search the neighboring individuals of the current individuals in the solution space, so as to find better neighboring solutions to increase the solution quality. This paper considers two local search methods: the row local search is to randomly swap two rows of the solution matrix (i.e., to swap two nurses’ schedules); the column local search is to rearrange the combination of shifts and days-off of all nurses in a random day (i.e., column).

The two local searches are illustrated in Figures 2(a) and 2(b). Figure 2(a) is the neighboring solution generated by row local search, in which the 2nd and the 4th rows are swapped. Figure 2(b) is the neighboring solution generated by column local search, in which the combination of work shifts and days-off of each nurse in the 4th column is rearranged.

4.5. Immigrant Scheme. The constraints of the concerned nurse scheduling problem result in a lot of infeasible solutions, which decrease the solution quality. Fortunately, the utilization of immigrant scheme can effectively decrease the number of infeasible solutions, which makes the GAIS able to handle large-scale complex problems efficiently. In executing the immigrant scheme, new immigrants (including random immigrants and elite immigrants) are generated first, and then those immigrants replace the worse or random individuals in the current population.

The random immigrants are the feasible solutions generated randomly, which aims to increase the diversification.
of the population. The elite immigrants are to reserve the better individuals in the current generation to the next generation. Those elite immigrants increase the ratio of those elite individuals in the population. To increase the solution convergence, a number of immigrants (where the number of elite immigrants is greater than the number of random immigrants) are used to replace the worse individuals in the current population, so that the number of those infeasible solutions could be decreased as the number of generations gets increased. Note that, in replacement of immigrants, new immigrants include a few random immigrants (which are generated randomly and aim to increase diversification of the whole population). The replacement of worse individuals in the current population can avoid the fact that the individuals with better fitness values in the current population are replaced by random immigrants with worse fitness values.

4.6. Stop Criteria. The purpose of the stop criteria is to limit the running time of the algorithm for searching a solution for our problem. The stop criteria determine the total executed number of iterations of the algorithm (i.e., three is a tradeoff for the stop criteria because a large number of iterations increase the running time but could improve the quality of the result at the same time). Our stop criteria are whether the maximal number of generations is achieved. The algorithm repeats its main loop until the stop criteria are achieved.

5. Performance Evaluation

In this section, the simulation experiments are conducted to evaluate the performance of our proposed algorithm for the concerned nurse scheduling problem. In our experiments, the GA without recovery scheme (called GA-I), the GA with recovery scheme (called GA-II), and the proposed GAIS are applied, respectively, to plan the 28-day nurse schedules with 20, 30, and 40 nurses. In addition, since the optimal solutions for all problems can be obtained by solving the mathematical programming model, our experimental results for GA-I, GA-II, and GAIS are also compared with the optimal solutions and the results using the preference satisfaction proposed in [5].

Before conducting the experiments, all the parameters are tested. The parameters on the nurse schedule include each nurse’s preferences for work shifts and days-off and the number of the assignments of the preferred work shifts and days-off during the previous planning period. The parameters on GA include the number of generations and the population size.

In this paper, all the experimental simulation is developed in C++ programming language; the applied optimization solver is IBM ILOG CPLEX Studio 12.4. All the experimental results are obtained by running on an Intel i7-3770 CPU 3.40 GHz with 16-GB RAM PC with a Windows 7 64-bit platform.

5.1. Parameter Setting. The parameter settings for GA-I, GA-II, and GAIS are listed in Table 2. In general, the number of generations and the population size would significantly

| Parameter/scheme | Value/method |
|------------------|--------------|
| Population size  | 160          |
| Population initialization | Generated randomly |
| Number of generations | 5000 |
| Selection scheme  | Roulette-wheel selection |
| Crossover rate    | 0.7          |
| Mutation rate     | 0.02         |
| Ratio of random immigrants | 0.2 |
| Ratio of elite immigrants | 0.4 |
| Replacing scheme  | The worst individuals are replaced |
| Stop criteria     | The maximal number of generations is achieved |
| Penalty scale $M$ | 2            |
influence the computational efficiency of GA and the solution quality. Although a GA with more generations and more individuals could obtain better solutions, the running time is increased drastically at the same time, and the computational effectiveness may not be good. To make our GAIS obtain good solutions in reasonable time, the number of generations and the population size would significantly influence the computational efficiency of GA and the solution quality. Although a GA with more generations and more individuals could obtain better solutions, the running time is increased drastically at the same time, and the computational effectiveness may not be good. To make our GAIS obtain good solutions in reasonable time, the number of generations and the population size used in our experiments are tested and analyzed first.

(i) Analysis on Number of Generations. The GAIS is applied to three problem instances (with 20, 30, and 40 nurses, resp.). Figure 3 gives the plot of fitness values versus the number of generations for the three problem instances.

From Figure 3, it can be observed that, during the first 1000 generations, the fitness values are increased rapidly; between 4000th and 5000th generations, the fitness values are improved slightly; after 5000th generation, the fitness values do not change a lot. Based on this, the number of generations in our later experiments is set to 5000.

(ii) Analysis on the Population Size. Figure 4 gives the plot of the average fitness values versus the population size for three different-scale problem instances with 9 different population sizes (including 20, 40, 60, 80, 100, 120, 140, and 160). From Figure 4, it can be observed that the increase of the population size really promotes the fitness value. When the population size is 160, the average fitness values are increased gradually. Therefore, the population size in our later experiments is set to 160, to make a balance between the algorithm efficiency and the solution quality.

5.2. Experimental Results. This subsection first uses the integer programming model in Section 2 to find the optimal solution and then compares the performance of GA-I, GA-II, and GAIS for three problem instances.

(i) The Optimal Solutions of Three Different-Size Problem Instances. The mathematical programming model developed in Section 2 is used to find the optimal solutions for three different-size problem instances, and the results are given in Table 3. Note that if the objective value approaches to 1, it means that almost all nurses’ preferences for work shifts and days-off are assigned. On the computational time, as the number of nurses is increased, the problem-solving time gets longer. As the number of nurses is increased to 100, it takes more than 7 hours but still cannot obtain any solutions. Therefore, it concludes that the mathematical programming model cannot efficiently solve the concerned nurse scheduling problem.

(ii) Comparison and Analysis of the Experimental Results Using GA-I, GA-II, and GAIS. In this experiment, we repeatedly run 20 times of GA-I, GA-II, and GAIS, respectively, for three problems. The performance measures are recorded in Table 4, including the best solution, the average solution, the worst solution, the standard deviation of running the 20 times of the algorithm, the deviation error, and running time.

The experimental results in Table 4 show that GA-II can produce feasible solutions for all problem scales; our GAIS can produce feasible solutions for small-scale problems (i.e., with less than 20 nurses) and middle-scale problems (i.e., with 20–40 nurses); GA-I can only produce feasible solutions for small-scale problems (i.e., with less than 20 nurses). That is, the solution quality of GA-II is the best, followed by our GAIS and followed by GA-I. Although the results show that our GAIS may not produce better-quality solutions than the GA-II for large-scale problems, our GAIS performs much efficiently compared to the GA-II (see Table 4). Additionally, most practical problems are of small scale or middle scale; it is hardly common to cope with large-scale problems.

First, the average and the best solutions of GA-II are very close; for example, in the third problem instance, the error deviation of the average solution from the best solution is about 6% (= (0.998 – 0.94)/0.998). However, as the number of nurses grows, the running time is increased drastically; for example, in the third problem instance, the average running time is greater than 30 minutes. Although the solutions of our GAIS perform slightly worse than the GA-II (e.g., in the third problem instance, the deviation error of the average solution from the best solution is less than the 0.2), the running time is much less (e.g., in the third problem instance, the average running time is saved by 30 minutes). Finally, although the running time of GA-I is the shortest, infeasible solutions in GA-I (i.e., the fitness values are negative) are often generated, so that the obtained average solutions (see Table 4) are worse.

In those experiments, both the GA-I and our GAIS allow for generating infeasible solutions, but as the number of generations is increased, the immigrant scheme of our GAIS

| Table 3: The optimal solutions of three different-size problem instances. |
|-----------------|-----|-----|-----|-----|
| # of nurses     | 20  | 30  | 40  | 100 |
| Optimal solution| 0.99| 0.982| 0.998| —   |
Table 4: Comparison of the experimental results using GA-I, GA-II, and GAIS.

| Problem   | Method | Best   | Average | Worst   | Std. Dev. | Deviation error | Time (s) |
|-----------|--------|--------|---------|---------|-----------|-----------------|----------|
| 20 nurses | GA-I   | 0.82   | 0.05    | −2.19   | 0.88      | 95%             | 29.94    |
|           | GA-II  | 0.98   | 0.97    | 0.96    | 0         | 0%              | 238.71   |
|           | GAIS   | 0.89   | 0.82    | 0.76    | 0.03      | 17%             | 72.34    |
| 30 nurses | GA-I   | 0.64   | −2.29   | −4.39   | 1.03      | 35%             | 95.55    |
|           | GA-II  | 0.98   | 0.97    | 0.96    | 0.01      | 1%              | 741.08   |
|           | GAIS   | 0.92   | 0.89    | 0.85    | 0.02      | 6%              | 116.52   |
| 40 nurses | GA-I   | −5.42  | −8.01   | −11.45  | 1.64      | 643%            | 83.06    |
|           | GA-II  | 0.94   | 0.94    | 0.93    | 0         | 6%              | 1988.92  |
|           | GAIS   | 0.87   | 0.84    | 0.81    | 0.02      | 14%             | 142.02   |
| 60 nurses | GA-I   | −18.48 | −21.94  | −25.45  | 2.06      | —               | 147.79   |
|           | GA-II  | 0.95   | 0.94    | 0.93    | 0.01      | —               | 3712.51  |
|           | GAIS   | 0.73   | −0.62   | −3.33   | 1.16      | —               | 214.69   |
| 80 nurses | GA-I   | −26.54 | −33.63  | −37.53  | 2.79      | —               | 193.88   |
|           | GA-II  | 0.92   | 0.91    | 0.90    | 0         | —               | 5229.75  |
|           | GAIS   | −9.41  | −13.23  | −19.42  | 290.89    | —               | 290.89   |
| 100 nurses| GA-I   | −24.55 | −33.90  | −46.56  | 5.72      | —               | 317.56   |
|           | GA-II  | 0.89   | 0.88    | 0.87    | 0.01      | —               | 10036.57 |
|           | GAIS   | −17.49 | −21.12  | −25.51  | 2.36      | —               | 410.11   |

Figure 4: Plot of the average fitness values versus the population size for different-scale problems.

In addition, the use of distributed sensors is useful to be applied to acquire the information on the number of available beds in hospital, which the schedule planner can be based on to determine the amount of the manpower of each work shift in each day, to avoid the fact that the work overloading decreases the satisfaction.

To solve the nurse scheduling problem, we first propose an integer programming model and then a genetic algorithm with immigrant scheme (GAIS) for different-scale problems. Such solutions are provided to decrease the time and pressure of the schedule planner. In particular, our proposed GAIS can effectively decrease the infeasible solutions to obtain better solutions for larger-scale problems.

In our experiments, our proposed algorithm is analyzed and compared with two standard GAs (with and without recovery scheme) for three problem instances. The experimental results show that our proposed algorithm can obtain a good-quality solution in a short time (about 2 minutes), which not only is feasible but also has smaller error deviation (about 20%).

In practical nurse preference scheduling, it is very important to generate a schedule in a short time. Nurses can immediately realize whether the scheduling result satisfies all their preferred work shifts and days-off. When a nurse realizes that not all her/his preferences are satisfied, she/he can immediately respond to the schedule planner to coordinate and readjust the scheduling results. Therefore, our proposed algorithm is reasonably capable of being applied to the practical nurse scheduling and other future applications.

6. Conclusion

This paper focuses on designing a novel preference satisfaction function of the nursing staff for nurse scheduling. In order to establish reasonable satisfaction, the preferences for work shifts and days-off are truly and fairly reflected in the nurse schedule. We not only consider the multiple preference ranks proposed in our previous work in [5] but also balance the numbers of the preferred work shifts and days-off, so that the planned schedules are not biased to the larger number of the preferences.

In practice, the use of distributed sensors is useful to be applied to acquire the information on the number of available beds in hospital, which the schedule planner can be based on to determine the amount of the manpower of each work shift in each day, to avoid the fact that the work overloading decreases the satisfaction.

To solve the nurse scheduling problem, we first propose an integer programming model and then a genetic algorithm with immigrant scheme (GAIS) for different-scale problems. Such solutions are provided to decrease the time and pressure of the schedule planner. In particular, our proposed GAIS can effectively decrease the infeasible solutions to obtain better solutions for larger-scale problems.

In our experiments, our proposed algorithm is analyzed and compared with two standard GAs (with and without recovery scheme) for three problem instances. The experimental results show that our proposed algorithm can obtain a good-quality solution in a short time (about 2 minutes), which not only is feasible but also has smaller error deviation (about 20%).

In practical nurse preference scheduling, it is very important to generate a schedule in a short time. Nurses can immediately realize whether the scheduling result satisfies all their preferred work shifts and days-off. When a nurse realizes that not all her/his preferences are satisfied, she/he can immediately respond to the schedule planner to coordinate and readjust the scheduling results. Therefore, our proposed algorithm is reasonably capable of being applied to the practical nurse scheduling and other future applications.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.
Acknowledgment

This work has been supported in part by NSC 101-2628-E-009-025-MY3 and MOST 103-2221-E-149-008, Taiwan.

References

[1] H. W. Purnomo and J. F. Bard, “Cyclic preference scheduling for nurses using branch and price,” Naval Research Logistics, vol. 54, no. 2, pp. 200–220, 2007.
[2] C. Valouxis and E. Housos, “Hybrid optimization techniques for the workshift and rest assignment of nursing personnel,” Artificial Intelligence in Medicine, vol. 20, no. 2, pp. 155–175, 2000.
[3] J. F. Bard and H. W. Purnomo, “Preference scheduling for nurses using column generation,” European Journal of Operational Research, vol. 164, no. 2, pp. 510–534, 2005.
[4] P. A. Clark, K. Leddy, M. Drain, and D. Kaldenberg, “State nursing shortages and patient satisfaction: more RN—Better patient experiences,” Journal of Nursing Care Quality, vol. 22, no. 2, pp. 119–127, 2007.
[5] C.-C. Lin, J.-R. Kang, W.-Y. Liu, and D.-J. Deng, “Modelling a nurse shift scheduling with multiple preference ranks for shifts and days-off,” Mathematical Problems in Engineering, vol. 2014, Article ID 937842, 10 pages, 2014.
[6] M. Moz and M. vaz Pato, “A genetic algorithm approach to a nurse rostering problem,” Computers & Operations Research, vol. 34, no. 3, pp. 667–691, 2007.
[7] E. Burke, P. de Causmaecker, and G. V. Bergh, “A hybrid tabu search algorithm for the nurse rostering problem,” in Simulated Evolution and Learning: Proceedings of the 2nd Asia-Pacific Conference on Simulated Evolution and Learning (SEAL ’98), Canberra, Australia, November 24–27, vol. 1585 of Lecture Notes in Computer Science, pp. 187–194, Springer, Berlin, Germany, 1999.
[8] E. Burke, P. Cowling, P. De Causmaecker, and G. V. Bergh, “A memetic approach to the nurse rostering problem,” Applied Intelligence, vol. 15, no. 3, pp. 199–214, 2001.
[9] T. Inoue, T. Furuhashi, H. Maeda, and M. Takaba, “A proposal of combined method of evolutionary algorithm and heuristics for nurse scheduling support system,” IEEE Transactions on Industrial Electronics, vol. 50, no. 5, pp. 833–838, 2003.
[10] U. Aickelin and K. A. Dowsland, “An indirect genetic algorithm for a nurse-scheduling problem,” Computers & Operations Research, vol. 31, no. 5, pp. 761–778, 2004.
[11] B. Maenhout and M. Vanhoucke, “An electromagnetic meta-heuristic for the nurse scheduling problem,” Journal of Heuristics, vol. 13, no. 4, pp. 359–385, 2007.
[12] R. Bai, E. K. Burke, G. Kendall, J. Li, and B. McComll, “A hybrid evolutionary approach to the nurse rostering problem,” IEEE Transactions on Evolutionary Computation, vol. 14, no. 4, pp. 580–590, 2010.
[13] N. Todorovic and S. Petrovic, “Bee colony optimization algorithm for nurse rostering,” IEEE Transactions on Systems, Man, and Cybernetics Part A: Systems and Humans, vol. 43, no. 2, pp. 467–473, 2013.
[14] M. Hadwan, M. Ayob, N. R. Sabar, and R. Qu, “A harmony search algorithm for nurse rostering problems,” Information Sciences, vol. 233, no. 1, pp. 126–140, 2013.
[15] S. Topaloglu and H. Selim, “Nurse scheduling using fuzzy modeling approach,” Fuzzy Sets and Systems, vol. 161, no. 11, pp. 1543–1563, 2010.
