System Design and Implementation for Solving the Resident Physician Scheduling Problem

LEI-MING SUN (Department of Plastic Surgery, Taipei Medical University Hospital, Taipei, Taiwan)
CHI-WAY WANG (Department of Computer Science and Information Engineering, National Taiwan University, Taipei, Taiwan)
MING-HUI JIN (Institute for Information Industry, Taipei, Taiwan)
LI LIU (Medical Informatics, Taipei Medical University Hospital, Taipei, Taiwan)
CHENG-YAN KAO (Institute for Information Industry, Taipei, Taiwan)

Abstract: - This paper at first details the resident physician scheduling problem which is important for the hospital. The difficulties of resident physician scheduling problem are how to satisfy the safe schedule constraint, the physician specification constraint and the fair schedule constraint simultaneously. To minimize the penalties violating the above constraints, our study has adopted the evolutionary approach. In addition to employ the ordinary genetic operators, we have proposed a new mutation method called dynamic mutation for solving this problem effectively. The experimental result showed that our proposed algorithm performed well in searching optimal schedules. At last, a physician scheduling system has been designed and implemented according to the proposed algorithm.

Key-Words: - Genetic Algorithm; Resident Physician Scheduling Problem

1 Introduction

The Resident Physician Scheduling Problem (RPSP) is a difficult task for the hospital. It concerns not only the preferences of resident physicians but also the patients' safety. The most important benefit of solving this problem is to keep physicians in a healthy state capable of taking care of patients.

The goal of this study is to design a system in searching the optimal rosters for the resident physicians. Because the attending physicians don't have to stay in house during call activities, we will not touch their schedules. On the contrary, the resident physicians may have to rotate around different wards monthly. For the schedule designer, either their attending or the chief resident, arranging a fair and reasonable schedule for every member is a tedious and difficult task. A fair and reasonable schedule should ensure: (1) All physicians can have nearly the same working hours. (2) The physicians can assign their off duty days. (3) No consecutive shifts exist.

The previous study [5] has achieved some success in this problem. However, despite the author had taken the main constraints into consideration and had proposed a smart solution for this, he had solved only some unrealistic problems which the representative was insufficient. Besides, their paper did not deal with the situation of a shift requiring more than one physician, which is common in the emergency room department.

This paper is organized as follows. We first introduce the requirements of scheduling the monthly rosters for the resident physicians and then formally present the resident physician scheduling problem in Section 2. The algorithm for solving this problem is presented in Section 3. Experiments for testing performance and quality of solutions of the proposed algorithm are presented in Section 4. Based on the proposed algorithm, a scheduling system is built and implemented. The scheduling system is shown in Section 5. The conclusion and future works are drawn in Section 6.

2 Problem Formulation

2.1 The Problem Description

Resident physicians have duty shifts except the regular daytime activities. Before 2003, there were few regulations or guidelines about duty hours in America. The ACGME (Accreditation Council on

Graduate Medical Education) approved the common duty hour's standards for programs in all specialties in February 2003. This regulation restricts duty hours to 80 hours per week, averaged over a 4-week period. However, in most other countries, there is still no formal regulation on residents' duty shifts.

Sun.	Mon.	Tues.	Wed.	Thu.	Fri.	Sat.
	1	2	2	2	3	3
	2	2	2	2	2	2
3	2	2	2	2	3	3
2	2	2	2	2	2	2
3	2	2	2	2	3	3
2	2	2	2	2	2	2
3	2	2	2	2	3	3
2	2	2	2	2	2	2
3	2	2	2			
2	2	2	2			

Fig. 1. The number of physicians required for each shift in January 2007

Sun.	Mon.	Tues.	Wed.	Thu.	Fri.	Sat.
	A	AH	AH	AF	DGH	ACD
	BE	BD	BD	BC	BH	BE
CFG	CF	AC	AH	CF	ACE	CEG
BD	DE	DG	BD	CD	DG	BD
BEF	CE	DE	EH	EH	BEG	BEF
BF	FH	AF	AF	DF	FG	AF
EGH	AG	AG	BG	DG	CFG	BDG
CH	BH	DH	FH	EH	AH	BH
EFH	AE	BC	EG			
CG	DG	AG	CF			

Fig. 2. Monthly roster example

Fig. 1 shows the number of physicians required for each shift in January 2007. Fig. 2 is an example of a monthly roster satisfying the request in Fig. 1. In this schedule, the notation of 'BE' in the second shift of January 1st means that the physicians B and E are scheduled to be on duty. This meets the number of physicians required for the second shift of January 1st (see Fig. 1). In this roster, 8 physicians, identified as A to H, are scheduled and each day has two shifts.

However, the physician scheduling problem is not as simple as it looks because physicians have their demands for a good roster. The most often demands are listed as follows:

1. This schedule should avoid consecutive shifts: For example, in Fig. 2, the physician F has to be on duty for 24 hours on Jan. 14 and for 36 hours from Jan. 19 to. 20. This will make him exhausted and absent-mind. Thereafter, he may give incorrect instructions to his patients. The incorrect instructions may be irredeemable.
2. This schedule should be fair: The physician B may complain that this schedule asks him to be on duty for 18 shifts and 9 of them are in weekends. Meanwhile, the physician C has only 15 shifts and only 5 of them are in weekends.

This requires the schedule designers to assign not only almost the same number of duty shifts to all physicians but also the same number of weekend shifts. It is a great challenge for the schedule designer. However, even if the schedule designers can overcome this challenge, it may be still unfair since the work load of each shift of the same day is not the same. For example, in most cases, the working load of a night shift is harder than that of a day shift. That is, the fairness cannot be measured only by the number of duty shifts.

To treat this problem objectively, some scheduler assigns each shift weight points standing for the working load of that shift. Fig. 3 shows an example of the weight point defined for the each shift in January 2007. Based on this weight point definition, we can achieve the true fair state for the physicians.

3. The schedule can be customized for each physician. According to experiences of schedule designers, the physicians may wish to adjust the roster since they have preferences for certain days. In other words, the schedule designer should not be autocratic. He should allow the physicians to specify their unavailable shifts, and the schedule should meet the specifications from the physicians.

Sun.	Mon.	Tues.	Wed.	Thu.	Fri.	Sat.
	2.5	1.0	1.0	1.0	1.0	1.5
	2.5	1.0	1.0	1.0	1.2	1.7
1.5	1.0	1.0	1.0	1.0	1.0	1.5
1.7	1.0	1.0	1.0	1.0	1.2	1.7
1.5	1.0	1.0	1.0	1.0	1.0	1.5
1.7	1.0	1.0	1.0	1.0	1.2	1.7
1.5	1.0	1.0	1.0	1.0	1.0	1.5
1.7	1.0	1.0	1.0	1.0	1.2	1.7
1.5	1.0	1.0	1.0			
1.7	1.0	1.0	1.0			

Fig. 3. The weight point defined for each shift

For convenience, the first requirement is named as the Safe Schedule Constraint (SSC), the second requirement as the Fair Schedule Constraint (FSC) and the third one as the Physician Specifying Constraint (PSC). In Section 2.2 we will formally detail these notations and the above constraints.

2.2 The Notations

To facilitate the description of the Physicians Scheduling Problem, we define the following notations.

D1. There are m physicians $P_1, \ldots, P_m$.

D2. A schedule contains n shifts.

D3. For each $1 \le i \le n$, the shift Si is a binary sequence $<S_i(1), S_i(2), \ldots, S_i(m)>$, where $S_i(j) = 1$ means the physician Pj should be on duty in the shift S_i and $S_i(j) = 0$ otherwise.

D4. For each $1 \le i \le n$, D_i is the number of physicians which should be on duty in the ith shift Si.

D5. For each $1 \le i \le n$, W_i is a non-negative real number which specifies the point (working load) of the ith shift S_i.

D6. For each $1 \le i, j \le n$ and $1 \le j \le m$, the schedule specification of shift Si from the physician P_j is defined as E_{ij}, where

$$E_{ij} = \begin{cases} 1 & P_j \text{ would like to be on duty in shift } S_i \\ -1 & P_j \text{ would not like to be on duty in shift } S_i \\ 0 & \text{Otherwise.} \end{cases}$$

2.3 Solution Format and Constraints

Based on the definitions in Section 2.2, a schedule can be defined as a shift sequence $<S_1, S_2, \ldots, S_n>$ containing n shifts. Since D_i is the number of physicians which should be on duty in the i^{th} shift S_i, the Physician Demand Constraint (PDC) can be formally stated as C1 below.

C1. For each $1 \le i \le n$, $D_i = \sum_{j=1}^{m} S_i(j)$

According to the regulations regarding resident duty hours issued by the ACGME of America [1 and 5], the physicians should not be on duty for more then 30 hours. Each hospital should specify the number of maximal consecutive duty shifts to the schedule designers to ensure that the number of consecutive duty shifts of each physician will not exceed the number of maximal consecutive duty shifts. Since the time definitions of the shifts are not uniform, we define MCDS to be the number of maximal consecutive duty shift. Based on this definition, the Safe Schedule Constraint (SCC) can be formally stated as C2 below.

C2. For each $1 \le i \le n - MCDS + 1$ and $1 \le j \le m$,

$$\prod_{k=0}^{MCDS-1} S_{i+k}(j) = 0$$

Although the constraint C2 ensures that the each physician will not be on duty for more than or equal to MCDS consecutive shits, the physicians still wish to minimize the number of successive duty shifts. The schedule designers should consider the soft constraint C3 below.

C3. For each $1 < h \le MCDS$, $1 \le i \le n - h + 1$ and $1 \le j \le m$,

$$\prod_{k=0}^{h-1} S_{i+k}(j) = 0$$

The regulation proposed by the ACGME also mentioned that the physicians should rest for sufficient time after their duty shift. Although this is not a hard constraint, the schedule designer should take this into consideration. The time definitions of the shifts are not uniform, therefore we define MRS to be the number of minimal rest shifts. Based on this definition, the schedule designers should consider the soft constraint C4 below.

C4. For each $1 \le i \le n - MRS$ and $1 \le j \le m$, $S_i(j) = 1$ implies

$$\sum_{k=1}^{MRS} S_{i+k}(j) = 0$$

In all the cases we learned, the Fair Schedule Constraint (FSC) is a soft constraint. Given the point of each shift, the FSC can be formally stated as C5 below.

C5. For each two physicians P_i and P_j,

$$\sum_{k=1}^{n} W_k S_k(i) = \sum_{k=1}^{n} W_k S_k(j)$$

The final constraint considered in this study is the Physician Specifying Constraint (PSC). In real cases, the schedule designers cannot ensure that they can find a feasible solution which meets the specifications from all the physicians. The schedule designers often treat the PSC as a soft constraint. In most cases the physicians accept the result which dissatisfies few of their specifications. Given the schedule specification, the PSC and be formally stated as C6 below.

C6. $\sum_{i=1}^{n}\sum_{j=1}^{m} E_{ij} \otimes S_i(j) = 0$, where

$$A \otimes B = \begin{cases} 1 & \text{if } A = 1 \text{ and } B = 0 \text{ or } A = -1 \text{ and } B = 1 \\ 0 & \text{otherwise} \end{cases}$$

2.4 Resident Physician Scheduling Problem

In Section 2.3 we formally propose two hard constraints C1 and C2, and four soft constraints C3–C6 for the Resident Physician Scheduling Problem. Since there may exist no feasible solution satisfying the six constraints, the RPSP only requests the schedule designers to search for the optimal feasible schedules which satisfies the hard constraints C1 and C2 while minimizing the penalties of violating the soft constraints C3-C6.

The penalty of violating the soft constraint C4 should be strictly decreasing with the number of rest shifts between two successive duty shifts. To formally state the penalty between to duty shifts, for every two shifts S_i and S_j satisfying $i < j$ and for each physician P_k, we define the shift penalty function SPF(i, j, k) as follow.

D7. $SPF(i, j, k) = S_i(k) \times S_j(k) \times (i + MRS + 1 - j)^2$ if $j - i \le MRS$ and $SPF(i, j, k) = 0$ otherwise.

Based on this definition, the cost C_{SSC} defined in equation (1) not only increases with the penalty of violating the soft constraint C4 but also increases with the penalty of violating the soft constraint C3. Therefore, we employ the cost C_{SSC} in equation (1) as the penalty of violating the soft constraint C3 and C4 of the Safe Schedule Constraint.

$$C_{SSC} = \sum_{k=1}^{m}\sum_{i=1}^{n-1}\sum_{j=i+1}^{\min\{i+MRS,n\}} SPF(i,j,k) \tag{1}$$

The penalty of violating the soft constraint C5 definitely increasing with the variance of the work loads of all the physicians, therefore, we apply the cost C_{FSC} defined by equation (2) to be the penalty of violating the soft constraint C5.

$$C_{FSC} = \sum_{j=1}^{m}\left|\frac{1}{m}\sum_{k=1}^{m}\sum_{i=1}^{n} W_i S_i(k) - \sum_{i}^{n} W_i S_i(j)\right| \tag{2}$$

Since $E_{ij} \otimes S_i(j) = 1$ if and only if the schedule dissatisfies the specification of shift S_i from the physician P_j, we define the penalty of violating the soft constraint C6 to be the cost C_{PSC} in equation (3) below.

$$C_{PSC} = \sum_{i=1}^{n} \sum_{j=1}^{m} E_{ij} \otimes S_i(j) \quad (3)$$

According to the definitions D1–D7 and the definitions of the cost C_{SSC}, C_{FSC} and C_{PSC} above, we propose the Resident Physician Scheduling Problem as follow.

Minimize

$$N_1(C_{SSC}) + N_2(C_{FSC}) + N_3(C_{PSC}) \quad (4)$$

Subject to

$$\forall\ 1 \le i \le n \text{ and } \forall\ 1 \le j \le m,\ S_i(j) \in \{0, 1\} \quad (5)$$

$$\forall 1 \le i \le n,\ D_i = \sum_{j=1}^{m} S_i(j) \quad (6)$$

$$\forall\ 1 \le i \le n - MCDS + 1 \text{ and } \forall\ 1 \le j \le m,$$

$$\prod_{k=0}^{MCDS-1} S_{i+k}(j) = 0, \quad (7)$$

Where

The three functions $N_1()$, $N_2()$ and $N_3()$ are increasing functions.

The detail description regarding the three increasing functions $N_1()$, $N_2()$ and $N_3()$ in the fitness function (4) for this study will be presented in Section 3.2.

3 The Proposed Genetic Algorithm

Fig. 4 shows the proposed GA for solving the RPSP. In Fig. 4, the algorithm first generates the initial population, and each chromosome in the initial population has a fitness value (See section 2.3, 2.4). Here, the higher fitness value means the worse chromosome, because of getting more values for violating constraints. The algorithm then applies the proposed reproduction method to generate an intermediate set of selected chromosomes called H1 which contains the same number of chromosomes as the initial population. Then the proposed GA applies the proposed crossover operator to generate two new chromosomes into the intermediate set H2. When the H2 contains the same number of chromosomes as the initial population, the algorithm applies the proposed mutation operator to generate the chromosomes in the next generation. At this stage, if the termination criteria are not satisfied, the algorithm continues the procedure until the termination criteria are satisfied.

In the proposed algorithm, the termination criteria first test whether the number of generations exceeds the user-defined value. If the number of generations does not exceed the user-defined value, the termination criteria then test whether there are any feasible chromosomes whose penalties are 0.

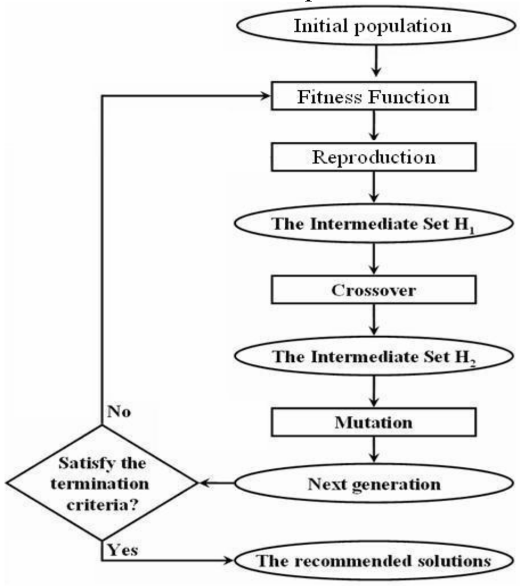

Fig. 4. The proposed genetic algorithm

3.1 The Chromosome and Initial Population

According to the definition of schedule in Section 2.3 and the definition D3 in Section 2.2, a chromosome is a binary sequence $\langle S_i(j)\rangle = \langle S_1(1)\ldots S_1(m), S_2(1)\ldots S_2(m), \ldots, S_n(1)\ldots S_n(m)\rangle$ where $S_i(j) \in \{0, 1\}$ for all i and j.

3.2 The Fitness Function

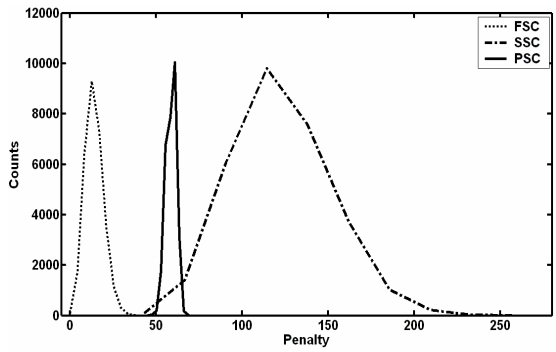

Fig. 5. The cost distributions

Fig. 5 shows the cost distributions of C_{FSC}, C_{PSC} and C_{SSC} from 30,000 chromosomes of the Problem in Section 4.1. All of the 30000 chromosomes are randomly generated. Their mean values are 16.249, 59.997and 134.32, and their standard deviation are 5.0729, 2.9176 and 28.522. This figure shows that the variance of distributions of the three costs is huge. To avoid that the nature of the cost distributions prejudiced the optimization algorithms against the costs C_{PSC} and C_{SSC}, the fitness function in this study first normalized the three costs before calculating the fitness of each chromosome. Specifically, the fitness function is this study is defined as $H_1Z(C_{SSC})$ +

$H_2Z(C_{SSC})$ + $H_3Z(C_{SSC})$, where Hi are positive constants for each i means the weight (importance) of the corresponding cost and

$$Z(x) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{x} e^{-y^2/2} \, dy$$

3.3 The Reproduction Methods

The proposed genetic algorithm employs two well-known reproduction methods to generate the intermediate set H1. The first one is the Roulette Wheel [3] method and the second one combine the Elitism [3] and Tournament [4] reproduction methods. The proposed algorithm allows the user to choose any one of them.

3.4 The Crossover and Mutation Operators

Given the mutation rate $R_C \in [0, 1]$. In this study, the proposed genetic algorithm takes the one-point crossover [2] and two-point crossover [2] operator into consideration. And given the mutation rate $R_M \in [0, 1]$, the proposed genetic algorithm takes the Bit Mutation [2]. The detail of those operators could be found in the references, but a revised mutation operator will be introduced below.

We reduce the mutation rate for the shifts which possess low probability to violate the PSC, if the physician meets the PSC. Based on this idea, we propose a new mutation operator called the dynamic mutation operator for the resident physician scheduling problem.

We need to take notice of the two hard constraints in the RPSP in Section 2.3, when we employ this algorithm.

4 The Experimental Results and Comparisons

To solve the resident physician scheduling problem, this study used two crossover operators, two mutation operators and a selection method with two parameters. In this section, one benchmark problem is introduced to test the performance and quality of solutions of each combination. The experiments in this study are performed on IBM PC with 2.0 GHz CPU and 2GB RAM running the Windows XP operating system. The values of the parameters H_1, H_2 and H_3 in the fitness function are all set as 1.

4.1 The Benchmark Problems

The specifications for this problem are presented below.

- There are 8 physicians. (m = 8)
- There are 31 days and each day has two shifts. The physicians' demands are specified in Fig. 1.
- The shift points are specified in Fig. 3.
- $\forall\, i \in \{1, 3, 5, 7, 9, 11\}$, $E_{i1} = 1$ and $E_{i2} = -1$
  $\forall\, i \in \{2, 4, 6, 8, 10, 12\}$, $E_{i2} = 1$ and $E_{i1} = -1$
  $\forall\, i \in \{13, 15, 17, 19, 21, 23, 25\}$, $E_{i3} = 1$ and $E_{i4} = -1$
  $\forall\, i \in \{14, 16, 18, 20, 22, 24, 26\}$, $E_{i4} = 1$ and $E_{i3} = -1$
  $\forall\, i \in \{27, 29, 31, 33, 35, 37, 39\}$, $E_{i5} = 1$ and $E_{i6} = -1$
  $\forall\, i \in \{28, 30, 32, 34, 36, 38, 40\}$, $E_{i6} = 1$ and $E_{i5} = -1$
  $\forall\, i \in \{41, 43, 45, 47, 49, 51, 53\}$, $E_{i7} = 1$ and $E_{i8} = -1$
  $\forall\, i \in \{42, 44, 46, 48, 50, 52, 54\}$, $E_{i8} = 1$ and $E_{i7} = -1$
  $E_{ij} = 0$ otherwise
- $MCDS = 4$.
- $MRS = 3$.

In current stage, we haven't found the global optimum solution for this problem. We propose this problem to test the quality of solutions (fitness) of each combination of the crossover and mutation operators. Besides, since we have experiences of the reproduction methods in GA, we've found that Elitism + Tournament is better than roulette wheel. Therefore, we choose E(0.3)+T(3) as the reproduction operator in this experience.

4.2 The Experimental Results and Comparisons

Fig. 6-9 show the best solution found by the proposed algorithm with four different combinations. We've noted that, first, no solutions in Fig. 6- 9 violate the physician specification constraint.

Sun.	Mon.	Tue.	Wed.	Thu.	Fri.	Sat.
	A	AH	AG	AF	AFG	ACF
	BH	BC	BE	BE	BD	BG
ACH	BC	CH	CG	AC	CEF	CEF
DH	DH	DF	DE	DH	BD	DE
CDE	BE	EG	DE	AE	BEH	CEG
DF	FG	CF	FG	BF	AF	FG
ABG	EG	CG	CG	EG	ADG	CDG
DH	AH	AH	FH	FH	AEH	CH
BDF	AB	AC	BG			
BE	AG	EF	DH			

Fig. 6. The best solution of single point crossover and bit mutation

Sun.	Mon.	Tue.	Wed.	Thu.	Fri.	Sat.
	A	AH	AE	AC	AEF	ACH
	BG	BD	BH	BE	BG	BE
CFH	CF	CG	AC	CG	BCG	ACG
DE	BD	DH	DF	DH	DG	DE
CEH	BE	DE	DE	AE	CEH	CEH
BF	BF	FH	AF	DF	AF	DF
BFG	CG	BG	DG	BG	ABG	DFG
FH	FH	EH	CH	AH	AH	CH
CDE	DE	BE	CD			
AG	BF	AG	AF			

Fig. 7. The best solution of single point crossover and dynamic mutation

Sun.	Mon.	Tue.	Wed.	Thu.	Fri.	Sat.
	A	AG	AC	AC	AFG	ACE
	BC	BG	BH	BF	BG	BD
CEH	CH	CE	CG	CH	CGH	CEF
BD	DG	AD	DF	DE	BD	AD
BEH	EG	CE	EG	BE	DEH	BDE
DF	FH	AF	CF	FH	CF	DF
BEG	FG	DG	AG	AG	AFG	AEG
EH	DH	BH	FH	AH	FH	CH
ABF	CD	AG	BD			
GH	EF	CF	AG			

Fig. 8. The best solution of two point crossover and bit mutation

Sun.	Mon.	Tue.	Wed.	Thu.	Fri.	Sat.
	A	AF	AC	AF	ADF	AGH
	BG	BH	BE	BE	BE	BG
CEG	CE	AC	CE	AC	CGH	ACE
DG	BD	DE	DF	DF	BD	DH
ABE	AE	EH	EH	AE	AEH	BCE
FH	DF	CF	DF	FG	BF	FH
CDG	AG	CG	CG	BG	BCG	BCG
FH	DH	AH	EH	AH	DH	FH
CDF	EG	CD	CF			
BD	AF	AE	DG			

Fig. 9. The best solution of two point crossover and dynamic mutation

The standard deviations of the weight of the 8 physician in the four solutions are 0.362284, 0.226385, 0.315945 and 0.392565, respectively. This also supports the previous conclusion that the combination of 'single-point crossover +dynamic mutation' surpasses others in minimizing the cost C_{FSC}. The mean times of violating the SSC of the 8 physician in the four solutions are 2, 1.5, 1.125 and 0.875, respectively. And it also supports the previous conclusion that the combination 'two-point crossover + dynamic mutation' performs better than others in minimizing the cost C_{SSC}. However, it should be noted that the differences of the standard deviations of the weight of the four solutions are small and the values are all acceptable for most physicians. On the other hand, the times of violating the SSC are much concerned by physicians. Therefore, we concluded that the quality of solutions of the solution in Fig. 9 is better than others and the combination 'Two-point crossover + dynamic mutation' is a better choice for the proposed genetic algorithm.

5 Scheduling System for the Physician Schedule Designer

The system is designed for hospitals, so there is an interface for managing all the faculties in every department.

After logging in, we begin to set some basic scheduling information (See Fig. 10) which includes:

[1] Which department to be scheduled
[2] The faculties included
[3] The minimum rest shift (2 in section 2.1)
[4] The number of shifts per day
[5] The beginning and end dates

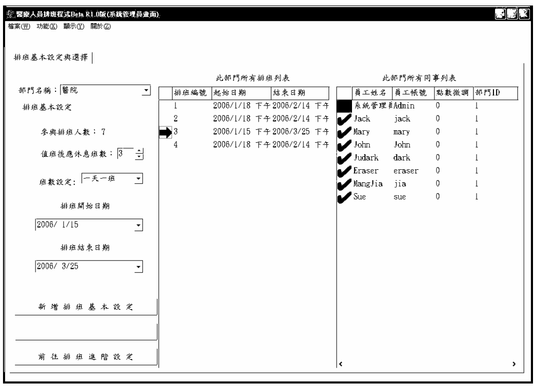

Fig. 10. Basic roster setting

Initially, we provide a template schedule. We usually have to continue the settings in order to satisfy the physicians' need.

[1] The number of faculty required for each shift (C1 in section 2.1)
[2] The weight point defined for each shift (1 in section 2.1)
[3] Faculty's demands (3 in section 2.1)

We could ask the system to produce different number of results at one time, and we also provide a chance for schedulers to change the parameters of the Genetic Algorithm to get different results. These parameters include:

- Generation times
- Population Size
- Crossover Rate & Mutation Rate
- Reproduction Method
- Crossover Method & Mutation Method

Finally, there may be numerous solutions for a scheduling problem. The system provides a summary list for making decision. It gives information of numbers of working days, weekend working days, weight points assigned, and the times of violating faculties' demands. Based on these, the scheduler can find a roster easier.

6 Conclusions and Future Work

In this paper, we've analyzed the requirements of scheduling the monthly rosters for the resident physicians and then formally presented the resident physician scheduling problem. To solve the resident physician scheduling problem, this study adopted the evolutionary approach and revised some operators. This study proposed a new mutation operator called dynamic mutation for minimizing the cost C_{PSC}. The experimental results showed that the combination 'two-point crossover + dynamic mutation' surpass other combinations and can produce a good result.

Nonetheless, this study is incomplete. We have no idea in determining the weights H_1, H_2 and H_3 for the fitness function. Currently, we assign them values arbitrarily. We are now trying to use machine learning approaches to automatically determine the weights.

References:

[1] Accreditation Council for Graduate Medical Education. *Report of the Work Group on Resident Duty Hours and the Learning Environment*, June 11, 2002.

[2] Davis, L. *Handbook of genetic algorithms*. Van Nostrand Reinhold, 1991

[3] Goldberg, D. E. *Genetic Algorithms in Search. Optimization, and Machine Learning*. Addison Wesley, 1989.

[4] Landon, W.B. *Genetic programming and data structures: Genetic Programming + Data Structures = Automatic Programming!*. The University of Birmingham. Kluwer Academic Publishers, 1998.

[5] Sun, L.M. *Using A Modified Genetic Algorithm to Solve the Scheduling of Physician's Shifts.* Master. Thesis, University of Taipei Medical, Taipei, Taiwan, June 2005.