

# A Genetic Algorithm for Resident Physician Scheduling Problem

Chi-Way Wang

Department of Computer Science  
and Information Engineering,  
National Taiwan University, Taiwan  
okeyawang@gmail.com

Lei-Ming Sun

Department of Plastic Surgery,  
Taipei Medical University Hospital,  
Taiwan  
miceray@gmail.com

Ming-Hui Jin

Institute for Information Industry,  
Taipei, Taiwan  
jinmh@iii.org.tw

Chung-Jung Fu

Taipei Medical University  
Wanfang Hospital, Taiwan  
subrina@wanfang.gov.tw

Li Liu

Medical Informatics,  
Taipei Medical University  
Hospital, Taiwan  
david@tmu.edu.tw

Chen-Hsiung Chan

Department of Computer  
Science and Information  
Engineering, National  
Taiwan University, Taiwan  
frankch@ntu.edu.tw

Cheng-Yan Kao

Department of Computer  
Science and Information  
Engineering, National  
Taiwan University, Taiwan  
cykao@csie.ntu.edu.tw

## ABSTRACT

This paper formally presents the resident physician scheduling problem, which is one of the most important scheduling problems in hospital. The resident physician scheduling problem is characterized as satisfying the fair schedule constraint, the physician specification constraint and the safe schedule constraint simultaneously. To minimize the penalties from violating the constraints, this study adopts the evolutionary approach to propose a genetic algorithm for solving the problems. In addition the well-known genetic operators, this study proposed a new mutation operator called dynamic mutation for solving the resident physician scheduling problem. The experimental results show that the proposed algorithm performs well in searching optimal schedules.

## Categories and Subject Descriptors

J.3 LIFE AND MEDICAL SCIENCES. - *Medical information systems*

## General Terms:

Algorithms, Management, Performance.

**Keywords:** Genetic Algorithm; Resident Physician Scheduling Problem

## 1. INTRODUCTION

The Resident Physician Scheduling Problem (RPSP) is a difficult task for hospitals. It concerns not only the duties of resident physicians but also the patients' safety. The benefits of solving this problem well are to keep physicians in a healthy condition capable of taking care of patients.

The goal of this study is to design an algorithm in searching the optimal rosters for the resident physicians. Because the attending physicians have their own shift duties, the resident physicians may

have to rotate around different wards monthly. For the schedule designer who may be the attending or the chief resident physician, arranging a fair and reasonable schedule for every member is often a 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 nurse scheduling problems [2, 3] are similar with the resident physician scheduling problem. Although referring to the problem formalization and algorithms for solving the nurse scheduling problems are helpful in solving the resident physician scheduling problem, the two problems are not the same. The main difference between the two problems arises from the degree of the physicians' and nurses' decision-making power. Most nurses may have less power in specifying their duties in certain shifts, but most physicians like to have power to specify their duty in every shift. The duty specification during each shift of the physicians is a constraint for the resident physician scheduling problem. This constraint is not conscientiously considered in most nurse scheduling problems.

The previous study in [7] focuses on solving the resident scheduling problem. Although the author in [7] presented the main constraints which are considered and proposed their method for solving the problem, they considered only some unrealistic problems and the formalization of RPSP is insufficient.

This paper is organized as follows. We analyze the requirements of scheduling the monthly rosters for the resident physicians and then formally present the resident physician scheduling problem in Section 2. Algorithms for solving the resident physician scheduling problem are presented in Section 3. Experiments for testing the performance and solution quality of the proposed algorithm are presented in Section 4. The conclusion and future works is drawn in Section 5.

## 2. RESIDENT PHYSICIAN SCHEDULING PROBLEM

### 2.1. The Problem Descriptions

Resident physicians have duty shifts except the regular daytime activities. Before 2003, there were few regulations or guidelines

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

GECCO '07, July 7–11, 2007, London, England, United Kingdom.

Copyright 2007 ACM 978-1-59593-697-4/07/0007...\$5.00.

about duty hours in America. The ACGME (Accreditation Council on Graduate Medical Education) approved the common duty hours 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. 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			

**Figure 1. The number of the physicians required for each shift in January 2007**

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

The physician scheduling problem is not 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 Figure 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-minded and he may give incorrect instructions to his patients thereafter. 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 during weekends. Meanwhile, the physician C has only 15 shifts and only 5 of them are in weekends.

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			

**Figure 2. A monthly roster example**

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.

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			

**Figure 3. The weight point defined for each shift**

To treat this problem objectively, some scheduler assigns each shift weight points specifying for the working load of that shift. Figure 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 the 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.

For convenience, the first requirement is named as the Fair Schedule Constraint (FSC), the second requirement as the Physician Specifying Constraint (PSC) and the third one as the Safe Schedule Constraint (SSC). 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  $P_1, \dots, P_m$ .
- D2. A schedule contains  $n$  shifts.
- D3. For each  $1 \leq i \leq n$ , the shift  $S_i$  is a binary sequence  $\langle S_i(1), S_i(2), \dots, S_i(m) \rangle$ , where  $S_i(j) = 1$  means the physician  $P_j$  should be on duty in the shift  $S_i$  and  $S_i(j) = 0$  otherwise.
- D4. For each  $1 \leq i \leq n$ ,  $D_i$  is the number of physicians which should be on duty in the  $i^{\text{th}}$  shift  $S_i$ .
- D5. For each  $1 \leq i \leq n$ ,  $W_i$  is a non-negative real number which specifies the point (working load) of the  $i^{\text{th}}$  shift  $S_i$ .
- D6. For each  $1 \leq i, j \leq n$  and  $1 \leq j \leq m$ , the schedule specification

of shift  $S_i$  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  $\langle S_1, S_2, \dots, S_n \rangle$  containing  $n$  shifts. Since  $D_i$  is the number of physicians which should be on duty in the  $i^{\text{th}}$  shift  $S_i$ , the Physician Demand Constraint (PDC) can be formally stated as C1 below.

C1. For each  $1 \leq i \leq n$ ,

$$D_i = \sum_{j=1}^m S_i(j)$$

According to the regulations regarding resident duty hours issued by the Accreditation Council on Graduate Medical Education (ACGME) of America [1 and 7], the physicians should not be on duty for more than 30 hours. Therefore, 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  $1 \leq i \leq n - MCDS + 1$  and  $1 \leq j \leq m$ ,

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

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

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

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

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. Since 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  $1 \leq i \leq n - MRS$  and  $1 \leq j \leq m$ ,  $S_i(j) = 1$  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 every 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. Therefore, the schedule designers 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 can be formally stated as C6 below.

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

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

According to the definitions of D3 and D6,  $E_{ij} = 1$  and  $S_i(j) = 0$  implies that the physician  $P_j$  would like to be on duty in the shift  $S_i$ , but the schedule dissatisfies this specification. Similarly,  $E_{ij} = -1$  and  $S_i(j) = 1$  implies that the physician  $P_j$  would not like to be on duty in the shift  $S_i$ , but the schedule dissatisfies this specification. Therefore,  $E_{ij} \otimes S_i(j) = 1$  if and only if the schedule dissatisfies the specification of shift  $S_i$  from the physician  $P_j$ .

### 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 not exist a feasible solution satisfying the six constraints, the Resident Physician Scheduling Problem 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 C5 definitely increases with the variance of the work loads of all the physicians, therefore, we apply the cost  $C_{FSC}$  defined by equation (1) to be the penalty of violating the soft constraint C5.

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

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 (2) below.

$$C_{PSC} = \sum_{i=1}^n \sum_{j=1}^m E_{ij} \otimes S_i(j) \quad (2)$$

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 two duty shifts, for each 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.

$$D7. \quad SPF(i, j, k) = \begin{cases} S_i(k) \times S_j(k) \times (i + MRS - j + 1)^2, & \text{if } j - i \leq MRS \\ 0 & \text{, otherwise} \end{cases}$$

Based on this definition, the cost  $C_{SSC}$  defined in equation (3) 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 (3) as the penalty of violating the soft constraint C3 and C4 of the Safe Schedule Constraint.

$$C_{SSC} = \sum_{k=1}^m \sum_{i=1}^{n-1} \sum_{j=i+1}^{\min\{i+MRS, n\}} SPF(i, j, k) \quad (3)$$

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

$$C_{FSC}, C_{PSC} \text{ and } C_{SSC} \quad (4)$$

Subject to

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

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

$$\forall 1 \leq i \leq n - MCDS + 1 \text{ and } \forall 1 \leq j \leq m, \quad (7)$$

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

### 3. THE PROPOSED GENETIC ALGORITHM

Figure 4 shows the proposed genetic algorithm for solving the Resident Physician Scheduling Problem. In Figure 4, the proposed algorithm first generates the initial population, and each chromosome in the initial population has a fitness value (See section 2.3, 2.4 and 3.2). 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  $H_1$  which contains the same number of chromosomes as the initial population. The proposed genetic algorithm then applies the proposed crossover operator to generate two new chromosomes into the intermediate set  $H_2$ . When the  $H_2$  contains the same number of chromosomes as the initial population, the algorithm then applies the proposed mutation operator to generate the chromosomes of the next generation. At this stage, if the termination criteria are satisfied, then the algorithm shows the recommended solutions and then terminates its process. If the termination criteria are not satisfied, then 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.

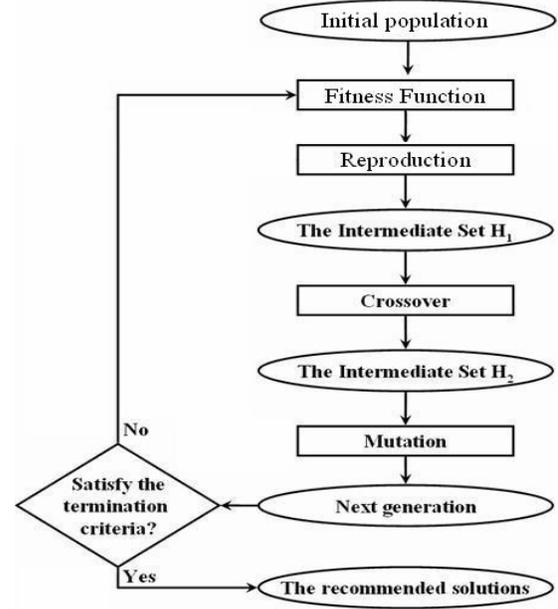


Figure 4. The proposed genetic algorithm

#### 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  $\langle S_i(j) \rangle = \langle S_1(1), \dots, S_1(m), S_2(1), \dots, S_2(m), \dots, S_n(1), \dots, S_n(m) \rangle$  where  $S_i(j) \in \{0, 1\}$  for all  $i$  and  $j$ .

The Resident Physician Scheduling Problem has two hard constraints. To ensure that the chromosomes do not violate the hard constraint C1 in Section 2.3, we employ the Algorithm I below to generate the chromosomes in the Initial Population. The chromosomes generated by Algorithm I automatically satisfy the hard constraint C1.

**Algorithm I:**

Step 1: Let  $i = 1$

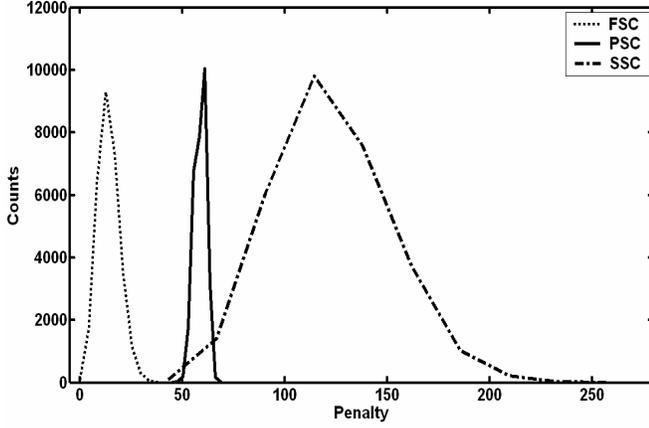
Step 2: Randomly select  $\pi_i \subseteq \{1, \dots, m\}$  containing  $D_i$  integers.

Step 3: Let  $S_i(j) = 1$  if  $j \in \pi_i$  and let  $S_i(j) = 0$  otherwise.

Step 4: If  $i = n$ , terminate this algorithm. Otherwise, let  $i = i + 1$  and then go to Step 2.

#### 3.2 The Fitness Function

Figure 5 shows the cost distributions of  $C_{FSC}$ ,  $C_{PSC}$  and  $C_{SSC}$  from 30,000 chromosomes of the Problem 1 in Section 4.1. All of the 30000 chromosomes are randomly generated by Algorithm I. Their mean values are 16.249, 59.997 and 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 the nature of the cost distributions prejudicing the optimization algorithms against the costs  $C_{PSC}$  and  $C_{SSC}$ , the fitness function



**Figure 5. The cost distributions.**

in this study first normalized the three costs before calculating the fitness of each chromosome. Specifically, the fitness function in this study is defined as  $H_1Z(C_{FSC}) + H_2Z(C_{PSC}) + H_3Z(C_{SSC})$ , where  $H_i$  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  $H_1$ . The first one is the Roulette Wheel [5] method and the second one combine the Elitism [5] and Tournament [6] reproduction methods. The proposed algorithm allows the user to choose any one of them.

If the user chooses the combination of Elitism and Tournament, he has to specify two parameters  $e$  and  $t$ . The parameter  $e \in [0, 1]$  specifies that the reproduction method to select the best  $e * L$  chromosomes from the current population, where  $L$  is the population size. The parameter  $t$  is a positive integer which specifies the reproduction method to randomly select  $t$  chromosomes from the current population and then choose the best one of them in each reproduction.

### 3.4 The Crossover Operators

In this study, the proposed genetic algorithm takes the one-point [4] and two-point crossover [4] operator into consideration with crossover rate  $R_C \in [0, 1]$ . Given two chromosomes  $\langle S_i(j) \rangle$ ,  $\langle T_i(j) \rangle$  and a real value  $z \in [0, 1]$ , if  $z < R_C$ , the one-point crossover operator first chooses an integer  $1 \leq x \leq n$  randomly and then generates two new chromosomes  $\langle A_i(j) \rangle$  and  $\langle B_i(j) \rangle$  whose definitions are stated below.

$$A_i(j) = \begin{cases} S_i(j) & \text{if } i \leq x \\ T_i(j) & \text{if } x < i \leq n \end{cases}$$

$$B_i(j) = \begin{cases} T_i(j) & \text{if } i \leq x \\ S_i(j) & \text{if } x < i \leq n \end{cases}$$

Given two chromosomes  $\langle S_i(j) \rangle$ ,  $\langle T_i(j) \rangle$  and a real value  $z \in [0, 1]$ , if  $z < R_C$ , the two-point crossover operator randomly chooses two integer  $1 \leq x \leq y \leq n$  and then generates two new chromosomes  $\langle A_i(j) \rangle$  and  $\langle B_i(j) \rangle$  whose definitions are stated below.

$$A_i(j) = \begin{cases} S_i(j) & \text{if } x < y, \text{ then } i \leq x \text{ or } i > y \\ & \text{if } x = y, \text{ then } i \leq x \\ T_i(j) & \text{if } x < y, \text{ then } x < i \leq y \\ & \text{if } x = y, \text{ then } x < i \leq n \end{cases}$$

$$B_i(j) = \begin{cases} T_i(j) & \text{if } x < y, \text{ then } i \leq x \text{ or } i > y \\ & \text{if } x = y, \text{ then } i \leq x \\ S_i(j) & \text{if } x < y, \text{ then } x < i \leq y \\ & \text{if } x = y, \text{ then } x < i \leq n \end{cases}$$

In this study, the proposed genetic algorithm applies the roulette wheel selection method to choose the chromosome  $\langle S_i(j) \rangle$  and chooses the chromosome  $\langle T_i(j) \rangle$  randomly at this stage.

### 3.5 The Mutation Operators

Given the mutation rate  $R_M \in [0, 1]$ , we employ the Algorithm II below to perform the Bit Mutation [4] operation to ensure that the chromosomes in the next generation do not violate the hard constraint  $C_1$  in Section 2.3.

**Algorithm II:**

- Step 1: Let  $i = 1$
- Step 2: Randomly select a real value  $x \in [0, 1]$ .
- Step 3: If  $x < R_M$ , then go to Step 5 to perform the mutation operation
- Step 4: If  $i = n$ , terminate this algorithm. Otherwise, let  $i = i + 1$  and then go to Step 2.
- Step 5: Randomly choose two distinct integers  $j$  and  $k$  from the set  $\{1, \dots, m\}$
- Step 6: Exchange the value of  $S_i(j)$  and  $S_i(k)$ , and then go to step 4.

Here we may 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. Given the mutation rate  $R_M \in [0, 1]$  and the restraint  $r \in [0, 1]$ , we employ the Algorithm III below to perform the dynamic mutation operation.

**Algorithm III:**

- Step 1: Let  $i = 1$
- Step 2: Randomly select an integer  $j$  from the set  $\{1, \dots, m\}$
- Step 3: If  $E_{ij} \otimes S_i(j) = 0$ , go to Step 9
- Step 4: Randomly select a real value  $x \in [0, 1]$ .
- Step 5: If  $x < R_M$ , then go to Step 7 to perform the mutation operation.
- Step 6: If  $i = n$ , terminate this algorithm. Otherwise, let  $i = i + 1$  and then go to Step 2.
- Step 7: Randomly choose a different integers  $h$  from  $j$  between the set  $\{1, \dots, m\}$
- Step 8: Exchange the value of  $S_i(j)$  and  $S_i(h)$ , and then go to step 6.
- Step 9: Randomly select a real value  $x \in [0, 1]$ . If  $x < r * R_M$ , then go to Step 7 to perform the mutation operation. Otherwise, go to Step 6.

## 4. EXPERIMENTAL RESULTS AND COMPARISONS

To solve the resident physician scheduling problem, this study considers two crossover operators, two mutation operators and a selection method with two parameters. In this section, two benchmark problems are introduced to test the performance and solution quality of each combination. All 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

#### Problem 1

The first benchmark problem and its optimal solution for this study are proposed by [7]. The specifications for this problem are presented below.

- There are 7 physicians. ( $m = 7$ )
- There are 70 days and each day has one shift. ( $n = 70$ )
- $\forall 1 \leq i \leq 70, D_i = 1$  and  $W_i = 1.0$
- $\forall 1 \leq i \leq 70$  and  $\forall 1 \leq j \leq 7, E_{ij} = 1$  if  $7 \mid (i - j)$  and  $E_{ij} = 0$  otherwise
- $MCDS = 4$ .
- $MRS = 3$ .

Sun.	Mon.	Tues.	Wed.	Thu.	Fri.	Sat.
$P_1$	$P_2$	$P_3$	$P_4$	$P_5$	$P_6$	$P_7$
$P_1$	$P_2$	$P_3$	$P_4$	$P_5$	$P_6$	$P_7$
$P_1$	$P_2$	$P_3$	$P_4$	$P_5$	$P_6$	$P_7$
$P_1$	$P_2$	$P_3$	$P_4$	$P_5$	$P_6$	$P_7$
$P_1$	$P_2$	$P_3$	$P_4$	$P_5$	$P_6$	$P_7$
$P_1$	$P_2$	$P_3$	$P_4$	$P_5$	$P_6$	$P_7$
$P_1$	$P_2$	$P_3$	$P_4$	$P_5$	$P_6$	$P_7$
$P_1$	$P_2$	$P_3$	$P_4$	$P_5$	$P_6$	$P_7$
$P_1$	$P_2$	$P_3$	$P_4$	$P_5$	$P_6$	$P_7$
$P_1$	$P_2$	$P_3$	$P_4$	$P_5$	$P_6$	$P_7$
$P_1$	$P_2$	$P_3$	$P_4$	$P_5$	$P_6$	$P_7$
$P_1$	$P_2$	$P_3$	$P_4$	$P_5$	$P_6$	$P_7$
$P_1$	$P_2$	$P_3$	$P_4$	$P_5$	$P_6$	$P_7$
$P_1$	$P_2$	$P_3$	$P_4$	$P_5$	$P_6$	$P_7$
$P_1$	$P_2$	$P_3$	$P_4$	$P_5$	$P_6$	$P_7$
$P_1$	$P_2$	$P_3$	$P_4$	$P_5$	$P_6$	$P_7$
$P_1$	$P_2$	$P_3$	$P_4$	$P_5$	$P_6$	$P_7$
$P_1$	$P_2$	$P_3$	$P_4$	$P_5$	$P_6$	$P_7$
$P_1$	$P_2$	$P_3$	$P_4$	$P_5$	$P_6$	$P_7$
$P_1$	$P_2$	$P_3$	$P_4$	$P_5$	$P_6$	$P_7$
$P_1$	$P_2$	$P_3$	$P_4$	$P_5$	$P_6$	$P_7$

Figure 6. The best roster for Problem 1

Figure 6 shows the only solution whose penalty is zero for this problem. In this problem, we recompute the average and standard deviation of  $C_{PSC}$  each generation, and we employ it to test the performance (generations for searching this solution) of each combination of the operators and parameters. The best combination will be applied in searching the optimization solution for the second benchmark problem.

#### Problem 2

The specifications for this problem are presented below.

- There are 8 physicians. ( $m = 8$ )
- There are 31 days and each day has two shifts. ( $n = 62$ )
- The physician demands are specified in Figure 1.
- The shift points are specified in Figure 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
- $MCDS = 4$ .
- $MRS = 3$ .

At current stage, we haven't found the optimal solution for this problem. In this study, we propose this problem to test the solution quality (fitness) of each combination of the crossover and mutation operators. The parameters of the selection method for minimizing

the cost of this problem are determined according to the best combinations in searching the optimal solution for Problem 1.

### 4.2. The Experimental Results and Comparisons

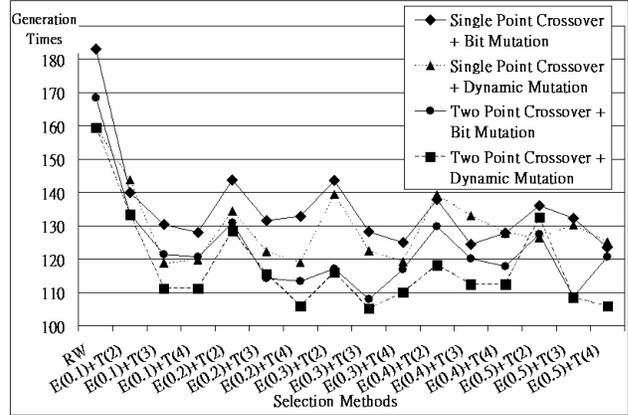


Figure 7. The experimental results for Problem 1

Figure 7 shows the experimental results which indicate the performance of each combination of crossover and mutation operators over difference parameters for the selection methods. The performance in this section means the average number of generations for finding the optimal solution in Figure 6. In Figure 7,  $RW$  in the  $x$ -axis means the selection method for the proposed algorithm is roulette wheel.  $E(e)+T(t)$  in the  $x$ -axis means the selection method for the proposed algorithm is a combination of elitism and tournament where the elitism parameter for the selection method is  $e$  and the tournament parameter for the selection method is  $t$ . The average value of each point is estimated from 30 samples.

In this experiment, the crossover rate is set as 1 for both the single-point and two-point crossover, and the mutation rate  $R_M$  is set as 0.005 for both the bit mutation and dynamic mutation operation. The restraint  $r$  for the dynamic mutation is set as 0.1, and 200 chromosomes in a generation.

Figure 7 supports the following conclusions. First, the performance of applying the combinations of elitism and tournament is much better than applying the roulette wheel for this problem. Second, the combination "two-point crossover + dynamic mutation" performs better than other combinations in most cases. Third, the best performance appears in the combination "two-point crossover + dynamic mutation" with the selection method  $E(0.3) + T(3)$ . Based on this conclusion, the second experiment applies the selection method  $E(0.3) + T(3)$  to compare the solution quality of the four combinations of the crossover and mutation operators while keeping all other parameters the same.

Figure 8 – Figure 10 show the distribution of the costs  $C_{FSC}$ ,  $C_{PSC}$  and  $C_{SSC}$  of the four combinations for problem 2 in the difference generations. Figure 8 shows that the combination "single-point crossover + dynamic mutation" performs better than others in minimizing the cost  $C_{FSC}$ . Figure 9 shows that all of the four combinations bring the same quality in minimizing the cost  $C_{PSC}$  after 100 generations, and Figure 10 shows that the combination "two-point crossover + dynamic mutation" performs better than others in minimizing the cost  $C_{SSC}$ . The three figures show that the costs convergent with no further improvement after 500 generations.

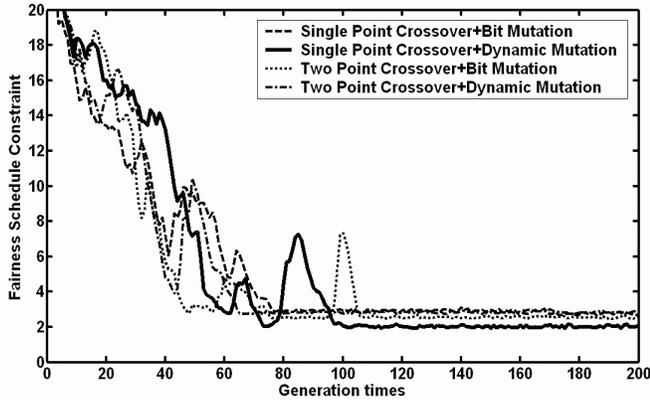


Figure 8. The distributions of the cost  $C_{FSC}$  for Problem 2

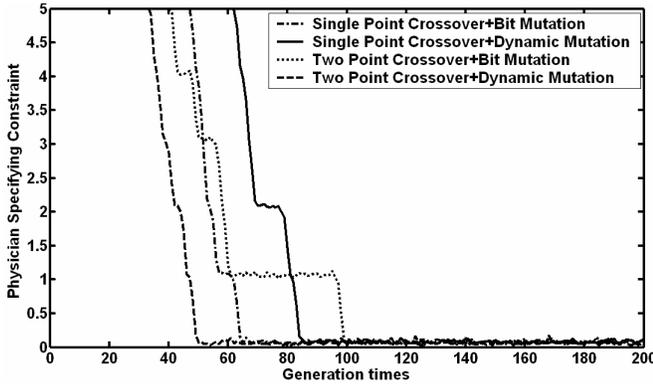


Figure 9. The distributions of the cost  $C_{PSC}$  for Problem 2

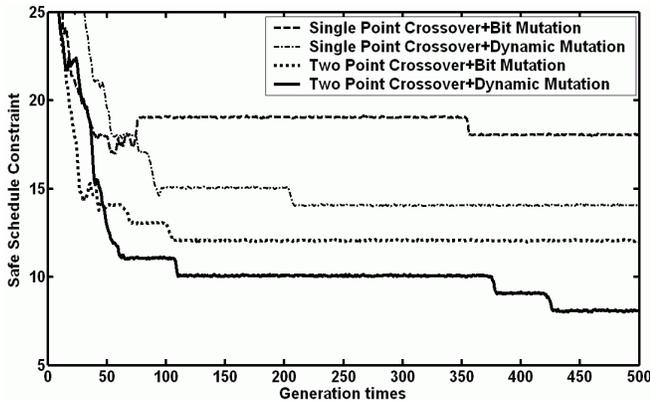


Figure 10. The distributions of the cost  $C_{SSC}$  for Problem 2

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			

Figure 11. The best solution of single point crossover and bit mutation for Problem 2

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			

Figure 12. The best solution of single point crossover and dynamic mutation for Problem 2

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			

Figure 13. The best solution of two point crossover and bit mutation for Problem 2

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			

Figure 14. The best solution of two point crossover and dynamic mutation for Problem 2

Figure 11 – Figure 14 show the best solution of problem 2 were found by the proposed algorithm with the four different combinations, and Figure 15 shows the weight and time of violating the safe schedule constraint of each physician in the solution of Figure 11 – Figure 14. Since the solutions in Figure 11 – Figure 14 violate no physician specification constraint, therefore, Figure 15 ignores the information regarding the PSC. In Figure 15,  $F_X$  means the solution in Figure X.

In Figure 15, the standard deviations of the weight of the 8 physician in the four solutions are 0.362284, 0.226385, 0.315945

and 0.392565. This also supports the previous conclusion that the combination “single-point crossover + dynamic mutation” performs better than others in minimizing the cost  $C_{FSC}$ . The means of the time of violating the SSC of the 8 physician in the four solutions are 2, 1.5, 1.125 and 0.875 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 time of violating the SSC is 1.5 would be less acceptable than the value 0.875. Therefore, we conclude that the solution quality of the solution in Figure 14 is better than others in Figure 11 – Figure 13 and hence we conclude that the combination “Two-point crossover + dynamic mutation” is better than other combinations for the proposed genetic algorithm.

Physician	Weight				Violate SSC times			
	F <sub>11</sub>	F <sub>12</sub>	F <sub>13</sub>	F <sub>14</sub>	F <sub>11</sub>	F <sub>12</sub>	F <sub>13</sub>	F <sub>14</sub>
A	20.2	20.6	20.7	21	2	3	2	0
B	20.3	20.6	20.3	20	3	3	0	1
C	20.7	20.7	20.9	21	1	0	0	0
D	20.7	20.3	20.2	20.5	2	0	2	1
E	21.1	20.6	20.2	20.2	2	2	1	1
F	20.1	20.8	20.8	20.5	0	2	1	1
G	20.9	20.1	20.9	20.9	3	2	2	3
H	20.3	20.6	20.3	20.2	3	0	1	0

Figure 15. Best solutions analysis

## 5. CONCLUSIONS AND FUTURE WORKS

In this paper, we analyze the requirements of scheduling the monthly rosters for the resident physicians and then formally present the resident physician scheduling problem. To solve the resident physician scheduling problem, this study adopts the evolutionary approach and takes several well-known operators and methods such as the single-point and two-point crossover operators, bit mutation, roulette wheel, elitism and tournament into

considerations. Besides, this study proposed a new mutation operator called dynamic mutation for minimizing the cost  $C_{PSC}$ . Several experiments were performed to compare the performance and solution quality of difference combinations. The experimental results shows that the combination “two-point crossover + dynamic mutation with the selection method  $E(0.3) + T(3)$ ” brings best performance and solution quality for solving the resident physician scheduling problem.

This study is incomplete. First, the design of a roster should take the information in previous monthly roster into consideration. In current stage, we can enlarge the chromosomes and enforce the algorithm to make no modifications to the boundary conditions (partial result in the previous roster). Second, we have no idea in determining the weights  $H_1$ ,  $H_2$  and  $H_3$  for the fitness function. Of current stage, manual determination by the experts is the only way determining the weights. We are now trying to use machine learning approaches to automatically determine the weights.

## 6. 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] Aickelin, U., E.K. Burke, and J. Li. An Estimation of Distribution Algorithm with Intelligent Local Search for Rule-based Nurse Rostering. *Journal of Operational Research Society*, pp. 1-12, 2006.
- [3] Bruke, E., P. De Causmaecker, S. Petrovic, and G.Vanden Berghe, Metaheuristics for Handling Time Interval Coverage Constraints in Nurse Scheduling. *Applied Artificial Intelligence*, pp. 743-766, 2006.
- [4] Davis, L. *Handbook of genetic algorithms*. Van Nostrand Reinhold, 1991.
- [5] Goldberg, D. E. *Genetic Algorithms in Search, Optimization, and Machine Learning*. Addison Wesley, 1989.
- [6] Landon, W.B. *Genetic programming and data structures : Genetic Programming + Data Structures = Automatic Programming!*. The University of Birmingham. Kluwer Academic Publishers, 1998.
- [7] 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.