PAPER

Nurse Scheduling by Cooperative GA with Effective Mutation Operator

Makoto OHKI[†a)], ***Member***

SUMMARY In this paper, we propose an effective mutation operators for Cooperative Genetic Algorithm (CGA) to be applied to a practical Nurse Scheduling Problem (NSP). The nurse scheduling is a very difficult task, because NSP is a complex combinatorial optimizing problem for which many requirements must be considered. In real hospitals, the schedule changes frequently. The changes of the shift schedule yields various problems, for example, a fall in the nursing level. We describe a technique of the reoptimization of the nurse schedule in response to a change. The conventional CGA is superior in ability for local search by means of its crossover operator, but often stagnates at the unfavorable situation because it is inferior to ability for global search. When the optimization stagnates for long generation cycle, a searching point, population in this case, would be caught in a wide local minimum area. To escape such local minimum area, small change in a population should be required. Based on such consideration, we propose a mutation operator activated depending on the optimization speed. When the optimization stagnates, in other words, when the optimization speed decreases, the mutation yields small changes in the population. Then the population is able to escape from a local minimum area by means of the mutation. However, this mutation operator requires two well-defined parameters. This means that user have to consider the value of these parameters carefully. To solve this problem, we propose a periodic mutation operator which has only one parameter to define itself. This simplified mutation operator is effective over a wide range of the parameter value.

key words: *nurse scheduling, cooperative genetic algorithm, mutation operator, optimization speed*

1. Introduction

General hospitals consist of several sections such as the internal medicine department and the pediatrics department. Each section is organized by about fifty to thirty nursing staffs. A section manager constitutes a roster, or a shift schedule, of all nurses of her/his section every month. In our interviewing research to the real hospitals, we found that the manager considers more than fifteen requirements for the scheduling. Such the schedule arrangement, in other words, the nurse scheduling, is a very complex task. In the interview, even a veteran manager can also spends 1 or 2 weeks to complete nurse scheduling. This means a great loss of work force. Therefore, computer software for the nurse scheduling has recently come to be required in the general hospitals [2]–[6], [8]–[17], [19], [21].

In an early study [2], the nurse scheduling problem, defined as a discrete planning problem, is solved by using Hopfield-model type-neural network. Berrada et al. [3] have proposed a technique to define the nurse scheduling problem as a multi-objective problem and to solve it by using a simple optimizing algorithm. The technique by Takaba et al. [4] provides a simple editing tool and simple GA for the nurse scheduling under Visual Basic environment. There are several techniques [5], [9], [16], [21] that require the user to modify or select the nurse schedule in the middle or the final stage of the optimization. Burke et al. apply a memetic approach to the nurse scheduling problem [6], [12], [14], [19]. Burke et al. [7] also define a technique to evaluate the nurse schedule. Some of these techniques are implemented in commercial nurse scheduling software. However, the evaluation technique does not fit to the shift system of our country. In our country, almost hospitals employ three-shift system. Therefore, we have defined the evaluation technique of the nurse schedule [18], [20], [22]. In the real wold, there are cases that nurses attend on a different day from the original schedule because of circumstances of other nurse or an emergency. There are also the cases that a nurse whom duty has been assigned originally takes a rest due to a disease. We discuss such a case that the nurse schedule has been changed in the past weeks of the current month. By such the changes, various inconveniences occur, for example, imbalance of the number of holidays and attendances. Such an inconvenience causes the fall of the nursing level of the whole nurse organization. Therefore, such inconvenience should be eliminated to aquire a better schedule. By considering the change of the shift schedule whenever one week passes, the shift schedule is reoptimized in remaining weeks of the current month.

In fact, the nurse schedule is still made by the hand of a manager or a chief nurse in many general hospitals. In our investigation, there are no general hospitals using such the commercial software for nurse scheduling. The manager is dissatisfactory to the the shift schedule generated by such the commercial software. And, many interactions to correct the schedule are also very complex for the user. The optimization algorithm of such the commercial software is still poor, and moreover, the schedule provided by such the software is hard to correct too.

In this paper, we discuss on generation and optimization of the nurse schedule by using the Cooperative Genetic Algorithm (CGA) [10]. CGA is a kind of Genetic Algorithm (GA) [1], and powerful optimizing algorithm for such a combinatorial optimization problem. In GA, individuals compete each other and superior individuals are preserved. On the other hand, individuals cooperate each other and the

Manuscript received August 31, 2011.
Manuscript revised February 2, 2012.
†The author is with the Division of Information and Electronics, Faculty of Engineering, Graduate School of Tottori University, Tottori-shi, 680–8552 Japan.
a) E-mail: mohki@ele.tottori-u.ac.jp
DOI: 10.1587/transinf.E95.D.1830

optimization of whole population progresses in CGA.

The conventional CGA optimizes the nurse schedule only by using crossover operator, because the crossover has been considered as the only one operation which keeps consistency of relation between chromosomes in the CGA, where the consistency means the number of nurses at each shift term in this case. In NSP treated in this paper, this consistency is positioned as a strong constraint. When CGA only with the crossover operator is applied to the nurse scheduling, the optimization often stagnates. Therefore we propose an effective mutation operator keeping the consistency for the CGA [22]. This mutation operator is activated depending on the optimization speed. However, this mutation operator requires two parameters to define itself. And also, these parameters are difficult to define, because several experiments and experiences are required. This means that the mutation operator depending on the optimization speed is effective but unfavorable for the user. To improve this problem, we propose a simple mutation operator activated periodically. We call this operator the periodic mutation operator.

2. Nurse Scheduling by CGA

2.1 Genetic Coding of the Nurse Schedule

In the nurse scheduling by CGA, an individual and its group, or the population, are defined shown in Fig. 1. The individual chromosome consists of the series of the shift symbols. The shift series consists of 28 fields, since almost hospitals handle four weeks as one month. The i-th individual expresses one-month schedule of the i-th nurse. In this problem, two or more individuals does not express the identical nurse's schedule. In other words, the population denotes the whole schedule.

There are several shift symbols to be put in the gene field as follows, symbols, D, S, M and H, denote a daytime shift, a semi-night shift, midnight shift and holiday respectively. Symbols, T, m, denote a training and a meeting,

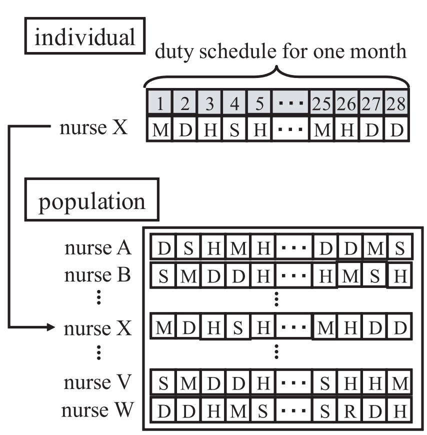


Fig. 1 The X-th individual coded into chromosome denotes one month shift schedule of the X-th nurse. The population includes one month schedules of all nurses.

where these are treated as a daytime shift term. Symbol, R, is a requested holiday which confirmed by the manager.

2.2 Evaluation of the Nurse Schedule

Interviews to sevral real general hospitals has been executed. By means of the interviews, a method to evaluate the nurse schedule is clarified. In this subsection, the evaluation of the nurse schedule is explaind.

For constituting the nurse schedule, the manager must consider many requirements. For example, the meeting, the training and the requested holiday must be accepted, where we assume that all the requested holidays have been confirmed by the manager. The semi-night shifts and the midnight shifts should be impartially arranged to all nurses. And arrangement of six or more consecutive shift days is prohibited. We have summarized all the requirements into the thirteen penalties. These penalties are classified into four groups.

2.2.1 Penalties on Work Load

We define a penalty function on the shift pattern as the following equation,

$$H_1 = \sum_{i=1}^{M} (h_{11}F_{1i} + h_{12}F_{2i} + h_{13}F_{3i}) \tag{1}$$

where F_{1i}, F_{2i} and F_{3i} denote the following penalty functions about the shift pattern and h_{11}, h_{12} and h_{13} denote the coefficients larger than or equeal to one.

We classify consecutive shift patterns for three days into four categories as shown in Table 1 (a). In this table, the meeting (m) and the training (T) are handled as the daytime shift, and the requested holiday (R) is handled as the holiday (H). The first category denotes a top priority pattern, and its penalty value is defined to zero. The second category denotes a priority pattern, and its penalty value is defined to one. The third category denotes a compromise pattern, and its penalty value is defined to two. The final category denotes a prohibited pattern, and its penalty value is defined to five. By comparing whole shift schedule of the i-th nurse with Table 1, the penalty is given by the following equation.

$$F_{1i} = \sum_{j=1}^{D-1} p_{ij} + p_{iD} \tag{2}$$

where p_{ij} denotes the penalty value given by Table 1 (a), p_{iD} denotes the penalty value defined to the final two day's shift pattern given by averaging all the penalty value of the consecutive two days appeared as shown in Table 1 (b) and D denotes the number of days of the object period.

The night shift should not be assinged to particular nurses intensively. To suppress this undesirable situation, we define the following penalty function to prohibit the X night shift or more for the consecutive Y days.

Table 1 Penalty values defined to consecutive three and two shifts.

(a) penalties defined to consecutive three days.

	p_{ij}	shift pattern				
advanced	0	DDD	DDH	DDM	DHD	DHH
		DHM	DMS	HDD	HDM	HHD
		HHH	HMS	SHH		
acceptable	1	DDS	DSH	DMH	HDH	HDS
		HHS	HSH	SHD	MHD	MHH
		MSH				
compromised	2	DHS	DSS	HHM	HSS	HMH
		SHS	SSH	MDH	MDS	MHS
		MMH				
unfavorable	5	DSD	DSM	DMD	DMM	HSD
		HSM	HMD	HMM	SDD	SDH
		SDS	SDM	SHM	SSD	SSS
		SSM	SMD	SMH	SMS	SMM
		MDD	MDM	MHM	MSD	MSS
		MSM	MMD	MMS	MMM	

(b) penalties defined to the final two days.

	D	S	M	H	p_{iD}
DD	0	1	0	0	0.13
DS	5	2	5	1	3.13
DM	5	0	5	1	2.87
DH	5	2	0	0	2.11
SD	5	5	5	5	5.00
SS	5	5	5	2	3.89
SM	5	5	5	5	5.00
SH	1	5	5	0	1.60
MD	5	2	5	2	3.50
MS	5	5	5	1	3.52
MM	5	5	5	2	3.89
MH	1	2	5	1	1.65
HD	0	1	0	1	0.50
HS	5	2	5	1	3.13
HM	5	0	5	2	3.23
HH	0	1	2	0	0.39

$$f_{2ij} = \begin{cases} \displaystyle\sum_{k=j}^{j+x-1} \frac{\max\left(N_{night/x}(i,k) - (y-1), 0\right)}{N_{night/x}(i,k)} & (j \in \textbf{SHIFT}) \\ 0 & (\text{otherwise}), \end{cases} \tag{3}$$

$$F_{2i} = \sum_{j=1}^{D} f_{2ij}, \tag{4}$$

where $N_{night/x}(i,k)$ denotes the number of the night shift assigned for consecutive x days starting from the $k-(x-1)$-th day in the shift schedule of the i-th nurse, y is defined as $y = xY/X$ and **SHIFT** denotes a set of days which shift is assigned.

In some hospitals, there are some cases to prohibit a specific shift pattern. If the shift pattern starting from the j-th day of the i-th nurse is prohibited, the penalty f_{3ij} is assigned to 1. We define a penalty function F_{3i} to implement such the prohibition as follows,

$$F_{3i} = \sum_{j=1}^{D} f_{3ij}. \tag{5}$$

2.2.2 Penalties on the Number of Shifts

We define a penalty function on the number of the shifts as the following equation,

$$H_2 = \sum_{i=1}^{M} (h_{21}F_{4i} + h_{22}F_{5i} + h_{23}F_{6i}) \tag{6}$$

where F_{4i}, F_{5i} and F_{6i} denote the following penalty functions about the number of the shifts and h_{21}, h_{22} and h_{23} denote the coefficients larger than or equal to one.

The number of the shifts should be impartially assigned to nurses. A total nursing level falls, if many shifts are concentrated to particular nurses. We define the following penalty functions to suppress unevenness of the number of shifts among nurses.

$$F_{4i} = \left|N_i^{hom} - N_{hom}\right|, \tag{7}$$

$$F_{5i} = \max\left(N_i^{sem} - N_{sem}, 0 \max\right) + \left(N_i^{mid} - N_{mid}, 0\right), \tag{8}$$

where N_i^{hom}, N_i^{sem} and N_i^{mid} denote the numbers of holidays, the semi-night shifts and the midnight shifts respectively assigned to i-th nurse. N_{hom} denotes the number of Saturdays and Sundays on the current month. N_{sem} and N_{mid} denotes the limit numbers of the semi-nightshift and the midnight shift, defined to four respectively in this paper.

If the shift is assigned to particular nurses on many consecutive days, total nursing level falls. We define the following penalty function to restrain assignment of the shift on consecutive shift days more than X.

$$f_{6ij} = \sum_{k=j}^{j+x-1} \frac{\max\left(N_{serial}(i,k) - (x-1), 0\right)}{N_{serial}(i,k)}, \tag{9}$$

$$F_{6i} = \sum_{j=1}^{D} f_{6ij}, \tag{10}$$

where $N_{serial}(i,k)$ denotes the number of consecutive shift days starting from $k-(x-1)$-th day in the shift schedule of i-th nurse and x is defined as follows,

$$x = \begin{cases} X & (j \le D - X) \\ y - 1 & (\text{otherwise}) \end{cases} \tag{11}$$

where y denotes the number of the remainder days of the optimization period.

2.2.3 Penalties on Nursing Level

We define a penalty function on the nursing level as the following equation,

$$H_3 = \sum_{j=1}^{D} \left(h_{31}F_{7j} + h_{32}F_{8j} + h_{33}F_{9j}\right), \tag{12}$$

where F_{7j}, F_{8j} and F_{9j} denote the following penalty functions about the nursing level and h_{31}, h_{32} and h_{33} denote the

Table 2 Nursing levels defined to twenty-three nurses in our practical example.

nurse	m_1	m_2	m_3	m_4	m_5	m_6	m_7	m_8
level	10	9	9	8	8	8	8	8
nurse	m_9	m_{10}	m_{11}	m_{12}	m_{13}	m_{14}	m_{15}	m_{16}
level	8	7	7	7	7	7	6	6
nurse	m_{17}	m_{18}	m_{19}	m_{20}	m_{21}	m_{22}	m_{23}	
level	5	5	4	4	3	2	1	

Table 3 Positions of twenty-three nurses.

nurse	m_1	m_2	m_3	m_4	m_5	m_6	m_7	m_8
position	chief	head	head	EX	EX	EX	EX	EX
nurse	m_9	m_{10}	m_{11}	m_{12}	m_{13}	m_{14}	m_{15}	m_{16}
position	EX	BB	BB	BB	BB	BB	BB	BB
nurse	m_{17}	m_{18}	m_{19}	m_{20}	m_{21}	m_{22}	m_{23}	
position	BB	BB	NF	NF	NF	NF	NF	

coefficients larger than one.

In our algorithm, the number of nurses in each working hours is preserved in any case. However, if new face nurses are intensively assigned on a particular working hours, the nursing level falls. The expert or more skilled nurses should be assigned for keeping nursing level. We assume that the manager evaluates the nursing level in ten phases. The nursing level of each nurse is given by ten phases as shown in Table 2 in our practical example. We define the following penalty functions to evaluate the nursing level of each working hours.

$$F_{7j} = \max\left\{L_j^{day} - \sum_i L(n_i), 0\right\}, \qquad n_i \in \mathbf{M}_j^{day}, \tag{13}$$

$$F_{8j} = \max\left\{L_j^{sem} - \sum_i L(n_i), 0\right\}, \qquad n_i \in \mathbf{M}_j^{sem}, \tag{14}$$

$$F_{9j} = \max\left\{L_j^{mid} - \sum_i L(n_i), 0\right\}, \qquad n_i \in \mathbf{M}_j^{mid}, \tag{15}$$

where L_j^{day}, L_j^{sem} and L_j^{mid} denote the lowest nursing level at each working term on the j-th day respectively, and $\mathbf{M}_j^{day}$, $\mathbf{M}_j^{sem}$ and $\mathbf{M}_j^{mid}$ denote the sets of nurses assigned at the daytime shift, the semi-night shift and the midnight shift on the j-th day respectively. In our practical example, L_j^{day} is defined as 54 on a weekday, 33 on Saturday and 28 on Sunday, and L_j^{sem} and L_j^{mid} are both defined to 16.

2.2.4 Penalties on Nurse Combination

We define a penalty function with respect to the nurse combination as the following equation,

$$H_4 = \sum_{j=1}^{D}\left(h_{41}F_{10j} + h_{42}F_{11j} + h_{43}F_{12j}\right), \tag{16}$$

where F_{10j}, F_{11j} and F_{12j} denote the following penalty functions with respect to the nurse combination and h_{41}, h_{42} and h_{43} denote the coefficients larger than one.

The manager also considers affinity between the nurses. Because of bad affinity between a certain nurses assigned to in the same time, there is the case that the nursing level deteriorates remarkably. We define a penalty function F_{10j}. When a pair of such the bad affinity is found in the shift schedule, penalty value 1 is added to the penalty function F_{10j}.

In the midnight shift, the number of assigned nurses is small. If the most of the nurses assigned to the midnight shift are new face nurse, the nursing level at the midnight shift falls remarkably To restrain such the unfavorable situation, we define the following penalty function,

$$F_{11j} = \begin{cases} 0 & N_{j,new}^{mid} < 2, \\ \displaystyle\sum_{i=0}^{N_{j,new}^{mid}-2}\left(N_{j,new}^{mid} - i - 1\right) & N_{j,new}^{mid} \geq 2, \end{cases} \tag{17}$$

where $N_{j,new}^{mid}$ denotes the number of new face nurses assigned to the night shift on the j-th day. In our practical example, we define positions of nurse as shown by the Table 3. In this table, EX, BB and NF denote an expert, a backbone and a new face respectively.

In general, one or more expert or more skilled nurses should be assigned to the daytime shift and the midnight shift. To restrain such an unfavorable situation, we define a penalty function F_{12j}. If no expert or more skilled nurse is assigned to the daytime shift and the midnight shift on the j-th day, the function F_{12j} is increased with one point.

2.2.5 Penalty on Reoptimization

At the real hospital, the shift schedule which optimized before the beginning of the current month is often changed. The change of the schedule leads to the disproportion of the number of the shift days among the nurses. It causes the overwork of particular nurses, if such unexpected situation is ignored. This means the falls of the nursing level, but may also lead to medical accidents as well. To restrain such an unfavorable situation, we consider the reoptimization of the shift schedule of the remainder of the current month. First, we assume that we have had the optimized shift schedule at the beginning of the current month. When several weeks have passed, we suppose that the shift schedule has been changed. CGA is applied to reoptimize the shift schedule of the next four weeks including the remainder of the current month. On the other hand, with considering the circumstances of the nurses, the shift schedule should not be changed as much as possible. Therefore, we define a penalty function F_{13} for reoptimizing the shift schedule while having such a dilemma. The penalty function F_{13} performs the difference between the original schedule and the newly optimized schedule of the remainder of the current month as shown in Fig. 2.

Finally, we define a total penalty function E of the shift schedule as follows,

$$E = \sum_{k=1}^{4} H_k + h_5 F_{13}. \tag{18}$$

where h_5 is the coefficient larger than one. The smaller value of the total penalty E means the better shift schedule.

3. Cooperative Genetic Algorithm for NSP

The basic algorithm of the CGA is as shown in Fig. 3 [18], [20], [22]. CGA applies the crossover operator to the population and searches so that a penalty of the whole population becomes small. The crossover operator selects a pair of parent individual from the population. Two offspring pairs are reconstituted by the two-point crossover. Taking back these offspring pairs to the original position of the parents, a temporal population is reconstituted. The temporal population is evaluated by the total penalty function E. These procedures are applied to one hundred parent pairs selected from the population while one generation cycle. A population giving the best performance is selected for the next generation.

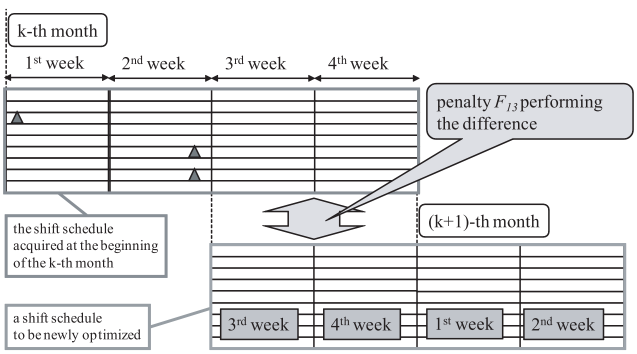


Fig. 2 We expand the nurse scheduling to accept some changes in the past two weeks. This figure shows an example when the two weeks have past, the coming four weeks are optimized to restrain inconvenience because of the changes.

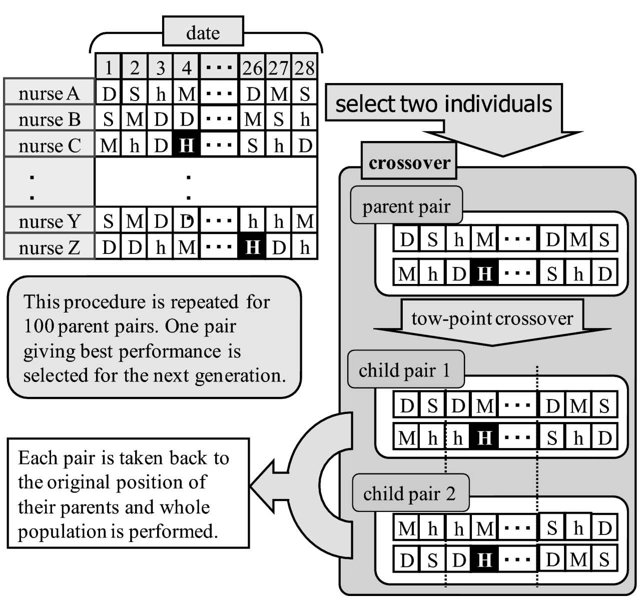


Fig. 3 one generation cycle by the crossover operator.

4. Mutation Operator Depending on Optimization Speed

Since NSP is particularly difficult to solve, the optimization which perform the crossover operator only often stagnates. The crossover operator is superior in ability to local search, but is inferior to global search. When the optimization stagnates while long generation term, it is effective to forcibly give small change to the population. The stagnation of the optimization can be observed by an optimization speed. Therefore we have proposed a mutation operator activated depending on the optimization speed [22].

The primitive operation of the mutation is shown in Fig. 4. The mutation operator randomly selects the day and selects two nurses. One of two nurses, giving the function F_1 big value, is stochastically selected. Another one is randomly selected. And then, these selected shifts are replaced each other.

The mutation operator is activated depending on the optimization speed as follows,

$$A(g) = \frac{1}{N_g} \sum_{i=0}^{N_g - 1} E(g - i) \tag{19}$$

$$V(g) = A(g-1) - A(g) \tag{20}$$

We define two parameters, a guard interval G_g and a speed threshold ϵ [22]. The guard interval is to prevent the activation of the mutation operator for G_g generation cycles after the last activation. The speed threshold is a parameter to detect activation timing. When the optimization speed become less than ϵ as shown in the following formula, the mutation operator activates,

$$V(g) < \epsilon. \tag{21}$$

We have tried computational experiment of the nurse scheduling by practical data. The number of the nurses is defined to twenty-three. The shift schedule sufficiently optimized has been announced at the beginning of the current month, and now assume that two weeks after. We suppose here that there have been several changes in the schedule in the past two weeks. The CGA reoptimizes the shift schedule for the coming four weeks. The schedule on the first two

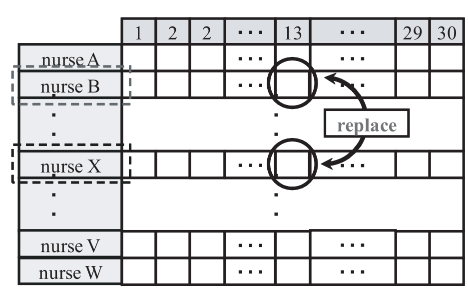
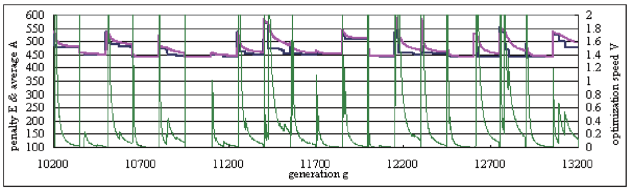
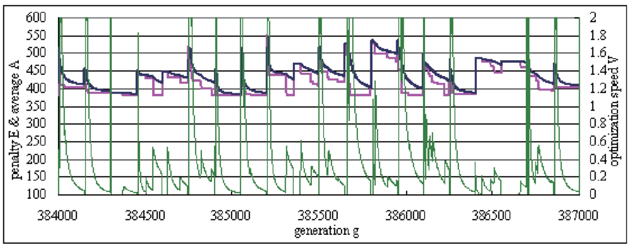


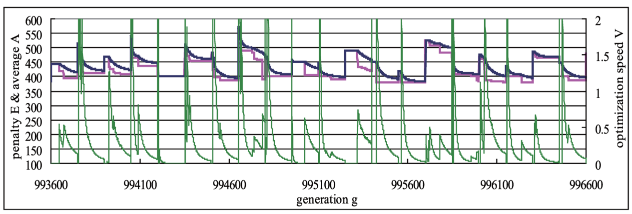
Fig. 4 Primitive operation of the mutation.



(a) 10200–13200



(b) 384000–387000



(c) 993600–996600

Fig. 5 The total penalty function, E(g), the average, A(g), and the optimization speed, V(g), in the beginning (a), the middle (b) and the end (c) of the optimization using the conventional mutation operator respectively.

weeks of the objective period has been already announced to the nurses at the beginning of the current month. In one trial, the optimization is executed for 1,000,000 generation cycles. In order to compare exactly, we have tried to optimize the ten times under each condition. The guard interval G_g is defined to 50 from our experience. As shown in Fig. 5, the mutation operator activated depending on the optimization speed. In this case, the guard interval G_g and the speed threshold ϵ are set to 50 and 0.01 respectively. As shown by Fig. 5, the mutation operator activated depending on the optimization speed. Thus, the optimization seems to be well progressed.

Figure 6 shows optimization results without the mutation operator and optimization results with the mutation operator depending on the optimization speed. Figure 7 shows optimization progresses by using the mutation operator depending on the optimization speed with guard interval $G_g = 50$ and speed threshold $\epsilon = 0.01$. This combination of the parameters has dedicated the best result in our experiment. The mutation operator depending on the optimization speed has worked effectively when the speed threshold ϵ has been defined to a value in the range from 0.01 to 0.2. When the speed threshold is defined to a value out of this range, the optimization results have been extremlly dissatisfactory.

The computing time is recorded in ten trials under the condition that the speed threshold and the guard interval is defined as 0.01 and 50 respectively. The minimum, the averaged and the maximum computing time is 8231 sec,

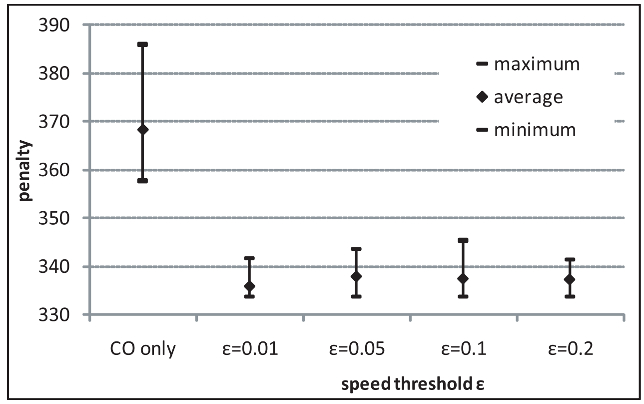


Fig. 6 Optimization results only with the crossover operator (CO only) and optimization results with the mutation depending on the optimization speed with several threshold values.

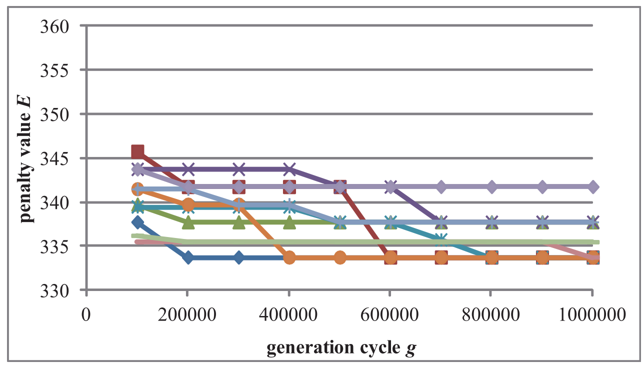


Fig. 7 Optimization progresses when the mutation depending on the optimization speed with guard interval $G_g = 50$ and speed threshold $\epsilon = 0.01$ is applied.

8480 sec and 10270 sec respectively.

5. Periodic Mutation Operator

The mutation operator with a threshold value, ϵ, except the range from 0.01 to 0.2 has brought unfavorable results. Besides, we must define the guard interval and the threshold appropriately. In other words, we have to be careful to handle the mutation depending on the optimization speed. So that, we propose a simple mutation operator activated periodically, where we call this operator the periodic mutation operator. The mutation operator is activated periodically every G_M generation cycles. Figure 8 shows the process flow of the optimization with the periodic mutation operator.

The periodic mutation operator requires only one parameter, the mutation period G_M, to define itself. As shown in Fig. 9, the periodic mutation gives results as almost equivalent to the conventional mutation operator. The mutation period is effective on wide range from 50 to 1000. This means the thing that does not have to mention a mutation period too much.

The computing time is recorded in ten trials under the condition that the mutation period is defined as 700. The

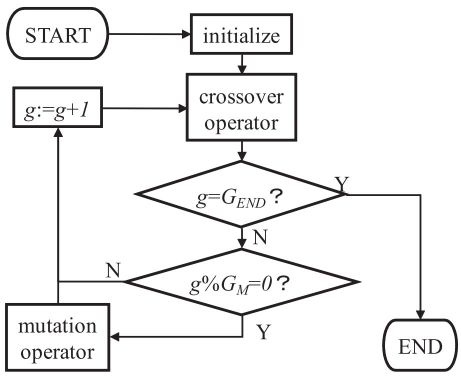


Fig. 8 process flow of the periodic mutation operator.

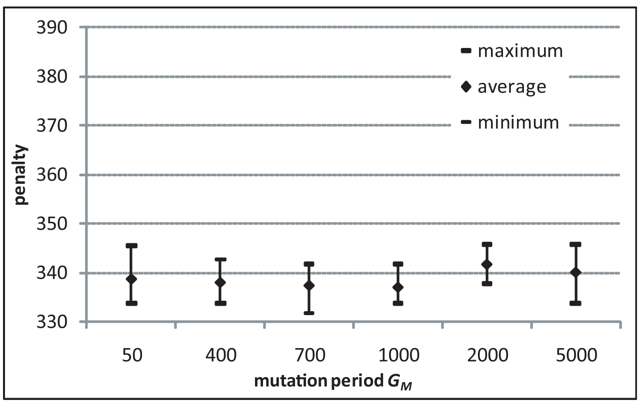


Fig. 9 Optimization results by the periodic mutation operator with several mutation periods. We have tried to set the mutation period from 50 to 5000.

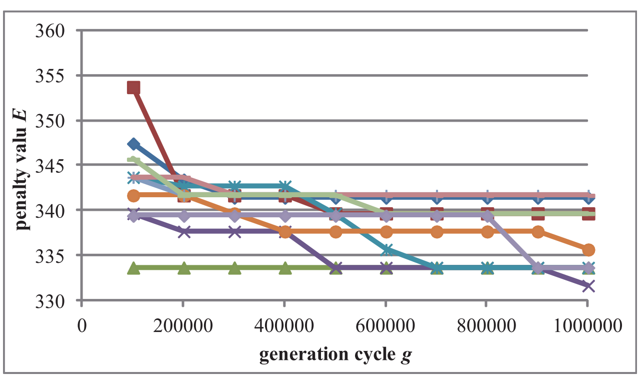
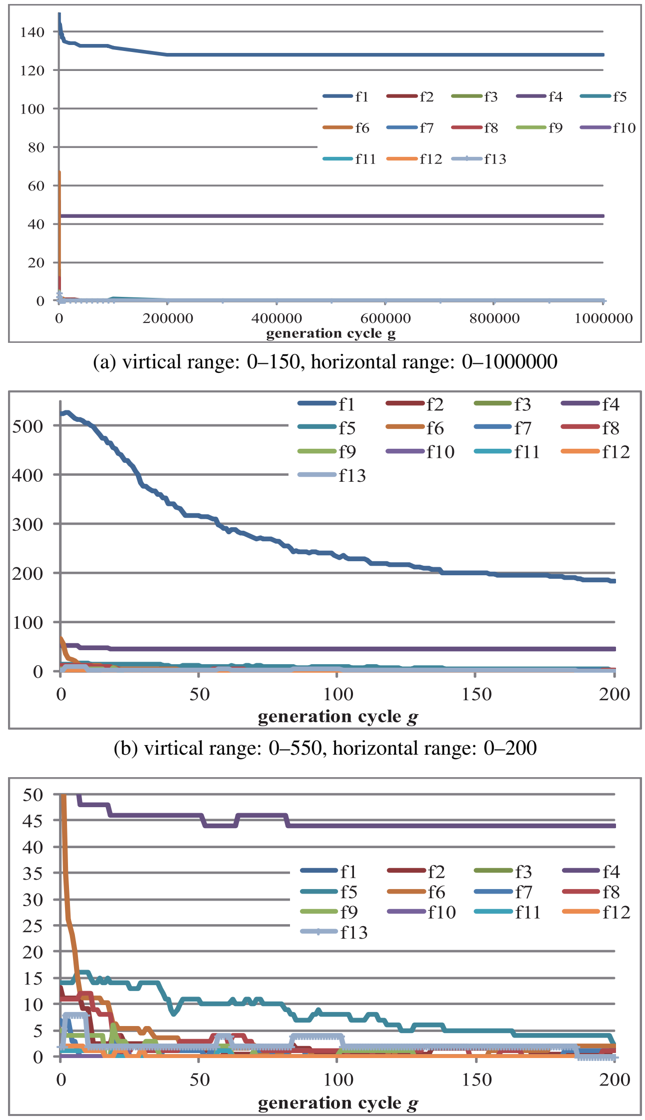


Fig. 10 Optimization progresses when the periodic mutation with mutation period G_M = 700 is applied.

minimum, the averaged and the maximum computing time is 8239 sec, 8356 sec and 9134 sec respectively. The computing time in the case when using the periodic mutation is almost same to when using the mutation depending on the optimization speed.

The optimization progresses are shown in Fig. 10. Especially under this condition which defines the mutation period 700, the splendid solution has been provided in the final stage of the optimization. As shown in Fig. 9, the periodic



(a) virtical range: 0–150, horizontal range: 0–1000000

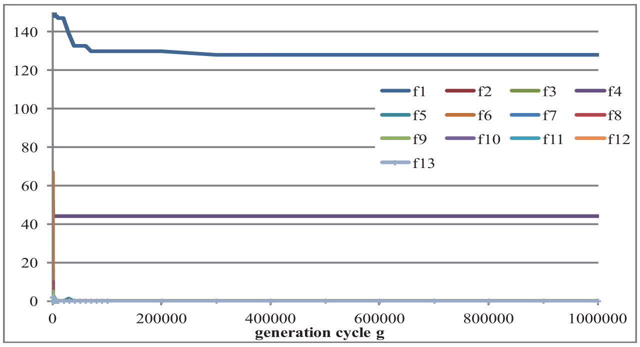
(b) virtical range: 0–550, horizontal range: 0–200

(c) virtical range: 0–50, horizontal range: 0–200

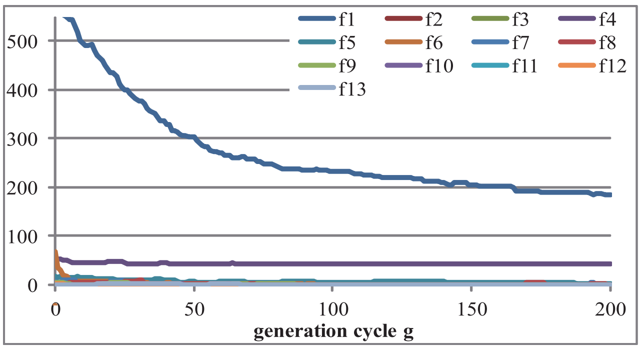
Fig. 11 An example of penalty progress when the mutation depending on the optimization speed with guard interval G_g = 50 and speed threshold ϵ = 0.01 is applied.

mutation is effective with wide range of the mutation period. However the complexity of the nurse scheduling problem varies under various conditions. It is considered that the value of the mutation period between 400 to 1000 is recommended for end users. This means that end users need not to mention about the value of the mutation period much. Therefore, the periodic mutation method is useful for end users.

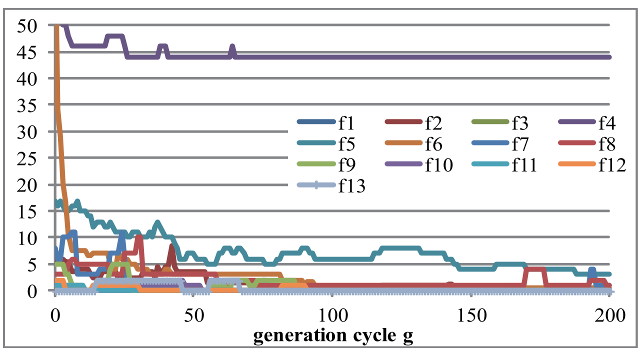
Figures 11 and 12 show penalty progress when the mutation depending on the optimization speed and the periodic mutation. This result shows that the search mechanism of both mutation methods are almost same. Surveying on the number of times of the mutation, for example, the number of times of the mutation has been 3240 in the case of the mutation depending on the optimization speed under a condition



(a) virtical range: 0–150, horizontal range: 0–1000000



(b) virtical range: 0–550, horizontal range: 0–200



(c) virtical range: 0–50, horizontal range: 0–200

Fig. 12 An example of penalty progress when the periodic mutation with mutation period $G_M = 700$ is applied.

that the speed threshold and the guard interval is defined as 0.01 and 50 respectively. On the other hand, the number of times of the mutation has been 1428 in the case of the periodic mutation with the mutation period, 700. These are one example. Regardless of the frequency of mutations, the value of the penalty F_{13} increases rapidly from zero in the early stages of the optimization, and then finally decreases gradually to zero.

6. Conclusion

This paper has introduced a nurse scheduling method that is based on CGA. We have discussed the case that the nurse schedule has been changed in the past weeks. To reoptimize the changed schedule, we have defined a penalty function to perform difference between the original schedule and the optimizing schedule.

Although CGA works to find a good schedule, the optimization with the crossover operator only often stagnates. To solve this difficulty, we have proposed the mutation operator activated depending on the optimization speed. Although this mutation operator searches effectively in the solution space of the nurse scheduling problem, several parameters have to be defined. To improve this inconvenience, we have proposed the periodic mutation operator. The periodic mutation offers the performance that is about the same as conventional mutation. Besides, the periodic mutation requires only one parameter, the mutation period. The mutation period can be defined in wide range of value.

In the future, a parallel processing technique for the nurse scheduling by using CGA should be considered. There are several aspects for the parallel processing, fine-grain parallelization and macro parallelization. In the idea of the fine-grain parallelization, when the mutation is activated, several mutated population can be generated. These are begun to be optimized in the mutation period in parallel. On the other hand, several cycles of the mutation period can be parallely executed. This is an aspect of the macro parallelization.

Acknowledgement

This research work is supported by Tottori University Electronic Display Research Center (TEDREC).

References

[1] D.E. Goldberg, Genetic Algorithm in Search, Optimization and Machine Learning, Addison-Wesley, New York, 1989.
[2] T. Goto, H. Aze, M. Yamagishi, M. Hirota, and S. Fujii, "Application of GA, neural network and AI to planning problems," NHK Technical report, no.144, pp.78–85, 1993.
[3] I. Berrada, J.A. Ferland, and P. Michelon, "A multi-objective approach to nurse scheduling with both hard and soft constraints," Socio-Econ. Plann. Sci., vol.30, no.3, pp.183–193, 1996.
[4] M. Takaba, H. Maeda, and N. Sakaba, "Development of a nurse scheduling system by a genetic algorithm," Proc. 18th JCMI, 1998.
[5] A. Ikegami, "Algorithms for nurse scheduling," Proc. 11th Intelligent System Symposium, pp.477–480, 2001.
[6] E.K. Burke and P. Cowling, "A memetic approach to the nurse rostering problem," Appl. Intell., vol.15, pp.199–214, 2001.
[7] E.K. Burke, P. De Causmaecker, S. Petrovic, and G.V. Berghe, "Fitness evaluation for nurse scheduling problems," Proc. 2001 Congress on Evolutionary Computation, 2001.
[8] H. Kawanaka, K. Yamamoto, T. Yoshikawa, T. Shinogi, and S. Tsuruoka, "Automatic generation of nurse scheduling table using genetic algorithm," IEEJ Trans. EIS, vol.122, no.6, pp.1023–1032, 2002.
[9] T. Inoue, T. Furuhashi, H. Maeda, and M. Takabane, "A study on interactive nurse scheduling support system using bacterial evolutionary algorithm enegine," IEEJ Trans. EIS, vol.122, no.10, pp.1803–1811, 2002.
[10] T. Itoga, N. Taniguchi, Y. Hoshino, and K. Kamei, "An improvement on search efficiency of cooperative GA and application on nurse scheduling problem," Proc. 12th Intelligent System Symposium, pp.146–149, 2003.
[11] B. Cheang, H. Li, A. Lim, and B. Rodrigues, "Nurse rostering prob-

lems - A bibliographic survey," Eur. J. Oper. Res., vol.151, pp.447–460, 2003.
[12] E.K. Burke, P. De Causmaecker, G.V. Berghe, and H. Lnadeghem, "The state of the art of nurse rostering," J. Scheduling 7, pp.441–499, 2004.
[13] A.T. Ernst, H. Jiang, M. Krishnamoorthy, B. Owens, and D. Sier, "An annotated bibliography of personnel scheduling and rostering," Annals of Operations Research, vol.127, pp.21–144, 2004.
[14] E.K. Burke, P. De Causmaecker, and G.V. Berge, "Novel metaheuristic approaches to nurse rostering problems in belgian hospitals," J. Leung Handbook of Scheduling Algorithms, Models and Performance Analysis, 2004.
[15] J. Li and U. Aickelin, "The application of bayesian optimization and classifier systems in nurse scheduling," Proc. 8th International Conference on Parallel Problem Solving from Nature, LNCS 3242, pp.581–590, UK, 2004.
[16] J.F. Bard and H.W. Purnomo, "Preference scheduling for nurses using column generation," Eur. J. Oper. Res., vol.164, pp.510–534, 2005.
[17] E. Oezcan, "Memetic algorithm for nurse rostering," Computer and Information Sciences - ISCIS 2005 Lecture Notes in Computer Science, vol.37332005, pp.482–492, 2005.
[18] M. Ohki, A. Morimoto, and K. Miyake, "Nurse scheduling by using cooperative GA with efficient mutation and mountain-climbing operators," 3rd Int. IEEE Conference Intelligent Systems, pp.164–169, 2006.
[19] E.K. Burke, P. De Causmaecker, S. Petrovic, and G.V. Berge, "Metaheuristics for handling time interval coverage constraints in nurse scheduling," Appl. Artif. Intell., vol.20, no.3, pp.1–19, 2006.
[20] M. Ohki, S. Uneme, S. Hayashi, and M. Ohkita, "Effective genetic operators of cooperative genetic algorithm for nurse scheduling," 4th Int. INSTICC Conference on Informatics in Control, Automation and Robotics, pp.347–350, 2007.
[21] J.F. Bard and H.W. Purnomo, "Cyclic preference scheduling of nurses using a lagrangian-based heuristic," J. Scheduling, vol.10, pp.5–23, 2007.
[22] S. Uneme, H. Kawano, and M. Ohki, "Nurse scheduling by cooperative GA with variable mutation operator," Proc. 10th ICEIS, INSTICC, pp.249–252, 2008.

Makoto Ohki In 1988, he joined the Electronic System Lab. Of OKI Electric Co., Ltd. Since 1994 he has been with Tottori University as a teaching and research assistant in the Department of Electrical and Electronic Engineering. At present he is an associate professor in the Division of Information and Electronic Engineering. His research interests are in application of GA, neural network and medical image processing. He is a member of IEEJ.