NURSE SCHEDULING BY COOPERATIVE GA WITH PENALTY COEFFICIENT ADJUSTMENT

Makoto Ohki and Hideaki Kinjo
Division of Information and Electronics, Graduate School of Tottori University
101, 4 Koyama Minami, Tottori, 680-8552 Japan

Keywords: Nurse scheduling, Genecit algorithm, Cooperative genecitic algorithm, Penalty coefficient adjustment.

Abstract: This paper describes a penalty adjustment technique for CGA applied to the nurse scheduling problem. The nurse scheduling is very complex task, because many requirements must be considered. In real hospital, some changes of the schedule often happen. Such a change of the shift schedule yields various inconveniences. Such an inconvenience causes the fall of the nursing level of the whole nurse organization. Furthermore, reoptimization of the schedule including such changes is very hard task and requires very long computing time. To improve this problem, we propose a technique to adjust penalty coefficient through the optimization.

1 INTRODUCTION

General hospitals consist of several sections such as the internal medicine department and the pediatrics department. Each section is arranged by the nursing staff of about fifty to thirty. A section manager makes a roster, or a shift schedule, of all nurses in her/his section every month. The manager considers more than fifteen requirements for the scheduling. Such the schedule arrangement, in other words, the nurse scheduling, is very complex task. Therefore, computer software for the nurse scheduling has recently come to be strongly required.

The shift schedule generated by such the commercial software is unsatisfactory. In fact, the nurse schedule is still made by the hand of the manager in many general hospitals. The optimization algorithm of such the commercial software is still poor. We discuss on generation and optimization of the nurse schedule by using the Cooperative Genetic Algorithm (CGA) (T. Itoga, 2003). CGA is a kind of Genetic Algorithm (GA) (D. E. Goldberg, 1989), and powerful optimizing algorithm for such a combinatorial optimization problem.

Burke et al. (E. K. Burke, 2001) have proposed a technique to evaluate the nurse schedule. However, this technique does not fit to the shift system of our country. Therefore, we have to define the evaluation technique of the nurse schedule. In the real case, there are some cases that nurses attend on a different day from the original schedule. We have discussed such a case that the schedule has been changed in the past weeks(M. Ohki, 2007; S. Uneme, 2008). The changed schedule must be reoptimized to avoid various inconveniences. Such an inconvenience causes the fall of the nursing level. Reoptimization of the schedule including such the changes is very hard task even by parallel computing techniques (M. Ohki, 2010b; M. Ohki, 2010a). We consider that this complexity is caused by that there are many local minima in the solution space of the nurse scheduling problem. We propose a technique adjusting penalty coefficient through the optimization when the concerned penalty function stagnate decreasing. If the optimization is caught in the region of the local minimum, some penalty functions stagnate decreasing. Valley of the local minimum upheaves by increasing weight of such the penalty function. And then, the searching point of the optimization escapes from the local minimum region.

2 NURSE SCHEDULING BY CGA

In the nurse scheduling by CGA, an individual and a population, are defined as shown in Fig.1. The individual consists of the series of the shift symbols. The shift series consists of 28 fields, where it means four weeks. The *i*-th individual expresses one-month schedule of the *i*-th nurse. In CGA, each individual denotes the schedule of each nurse. The population

Ohki M. and Kinjo H..
NURSE SCHEDULING BY COOPERATIVE GA WITH PENALTY COEFFICIENT ADJUSTMENT.
DOI: 10.5220/0003618902550258
In *Proceedings of the International Conference on Evolutionary Computation Theory and Applications* (ECTA-2011), pages 255-258
ISBN: 978-989-8425-83-6

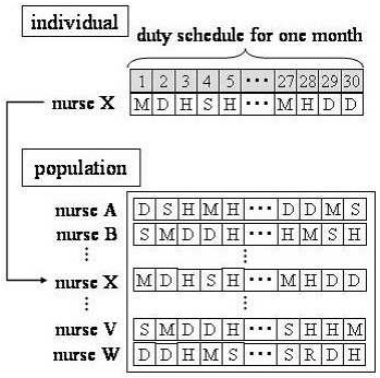


Figure 1: The individual and the population.

denotes one-month schedule.

We have summarized all the requirements into the 13 penalty functions to evaluate the population. Detail of these penalties has been shown in the previous research work (M. Ohki, 2010a). We describe an outline of each penalty.

The following penalty functions evaluate work load of each nurse. A penalty function F_{1i} evalutates the work loard of each nurse i for three consecutive days. A penalty function F_{2i} evaluates to prohibit the X night shift or more for the consecutive Y days. A penalty function F_{3i} evaluates specific prohibited shift pattern.

The following penalty functions evalute the number of the shifts impartially assigned. Penalty functions F_{4i}, F_{5i} supress unevenness of the number of shifts among nurses on the holidays and the night shifts. A penalty function F_{6i} restrains assignment of the shift on many consecutive shift days.

In our CGA, the number of nurses in each working hours is preserved in any case. We define penalty functions F_{7j}, F_{8j} and F_{9j} to evaluate the nursing level on the day time shift, the semi-night shift and the midnight shift respectively.

The following penalty functions evaluate about the combination of the nurses. A penalty function F_{10j} restrains unfavorable affinity of nurses. A penalty function F_{11j} restrains unfavorable situation that too many new faces are assigned on the midnight shift. A penalty function F_{12j} restrains unfavorable situation that one or more expert or more skilled nurses are not assigned to the daytime shift and the midnight shift.

A penalty function F_{13} performs the difference between the original schedule and the newly optimized schedule of the remainder of the current month to restrain the falls of the nursing level because of the change of the schedule.

Finaly, we perform the shift schedule by the following total penalty function at g-th generation,

$$
\begin{aligned}
E(g) = &\sum_i (h_1 F_{1i} + h_2 F_{2i} + h_3 F_{3i}) \\
&+ \sum_i (h_4 F_{4i} + h_5 F_{5i} + h_6 F_{6i}) \\
&+ \sum_j (h_7 F_{7j} + h_8 F_{8j} + h_9 F_{9j}) \\
&+ \sum_j (h_{10} F_{10j} + h_{11} F_{11j} + h_{12} F_{12j}) + h_{13} F_{13}, \quad (1)
\end{aligned}
$$

where h_1—h_{13} denote penalty coefficients.

The basic algorithm of the CGA is as shown in Fig.2 (M. Ohki, 2006; M. Ohki, 2007; S. Uneme, 2008). CGA applies the crossover operator to the population and searches so that a penalty of the whole population becomes small.

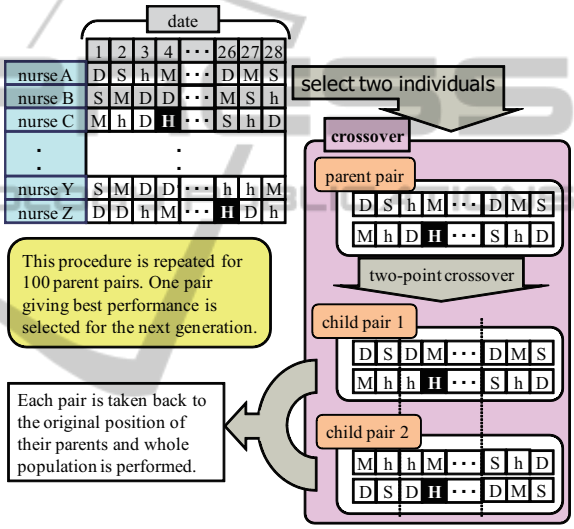


Figure 2: One generation cycle by the crossover operator.

When the optimization stagnates for long generation cycles, it is effective to forcibly give small change to the population. Therefore we have proposed a mutation operator activated depending on the optimization speed $V_E(g)$. When V_E becomes less than or equeal to a speedo-thleshold ε_E, the mutation is activated. We have defined two parameters, a guard interval G_g to prevent the activation of the mutation operator for G_g generation cycles right after the last activation.

We also have proposed a periodic mutation operator activated periodically in G_M generation cycles(M. Ohki, 2010b). This mutation is advantage on the point that fewer parameter is required to define itself.

3 PENALTY ADJUSTMENT

The shape of the solution space is defined by the pen-

alty function $E(g)$. By changing the penalty coefficients, the shape of the solution space is also changed. Valley of the local minimum upheaves by increasing weight of such the penalty function. And then, the searching point of the optimization escapes from the local minimum region. The optimization flow with Penalty coefficient Adjustment (PA) is as shown in Figure 3.

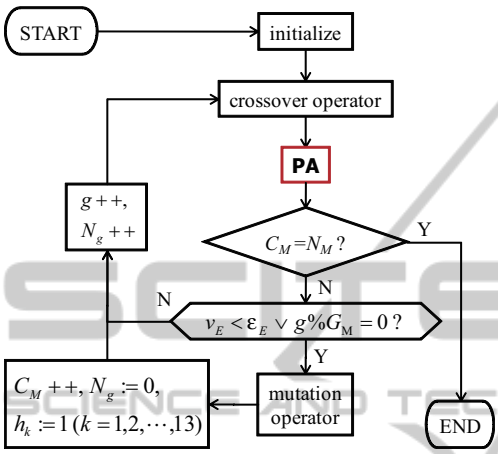


Figure 3: Optimization flow with the penalty adjustment.

First, all the penalty coefficients h_1—h_{13} are initialized to 1. The decreasing speed of each penalty function at the generation cycle g, $V_k(g)$, is calculated in PA by the following equations,

$$A_k(g) = \begin{cases} \dfrac{1}{N_g}\displaystyle\sum_{h=0}^{N_g-1}\sum_{i=1}^{M} F_{ki}(g-h) & (k \le 6), \\ \dfrac{1}{N_g}\displaystyle\sum_{h=0}^{N_g-1}\sum_{j=1}^{D} F_{kj}(g-h) & (k > 6), \end{cases} \tag{2}$$

$$V_k(g) = A_k(g-1) - A_k(g). \tag{3}$$

As shown in Figure 4, when the decreasing speed of the k-th penalty function, V_k, becomes less than or equal to a speed threshold ε_F, the penalty coefficient h_k is increased by multiplying with parameter α. In this research, the values of ε_F and α are defined as 0.01 and 1.01 respectively. When the mutation is activated, all the penalty coefficients h_1—h_{13} are initialized to 1 again.

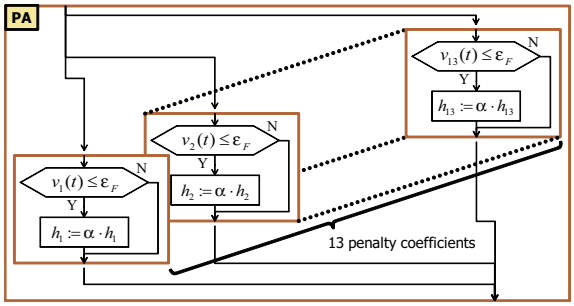


Figure 4: Primitive operation of the penalty coefficient adjustment.

of the variation of the penalty coefficients modified by PA. In the initinal stage of the optimization, several coefficients are increased extremely. While the optimization progresses, the penalty coefficients wiggle for long term of the generation cycle. In these generation cycles, the mutation is activated in a short period. In other words, the optimization stagnates frequently. This means that the optimization is caught in the local minimum region. Approaching the end of the optimization, we can find several generation intervals which the mutation is not activated and then some of coefficients are increased extremely. In these intervals, the optimization progresses for a long generation interval. In the final stage of the optimization, the progression of the optimization clearly represented. This suggests that the optimization goes towards convergence.

Figure 6 shows optimization progress by using the periodic mutation operator and PA respectively. The red dotted lines and blue solid lines denote maximum, average and minimum value of the penalty function E given by the periodic mutation and PA respectively. In one trial, the optimization is executed for $N_M = 500$ mutation cycles. By means of PA, the optimization finishes in about one-tenth generation cycles by the periodic mutation.

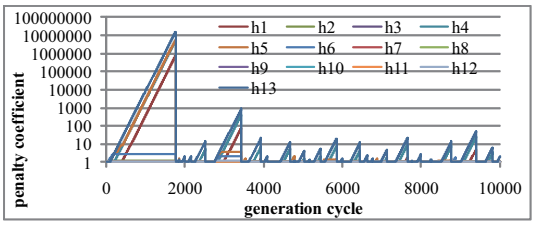
Figure 7 shows comparison of the maximum, the average and the minimum value of the ten results under each technique. Compared to the periodic mutation operator, PA is slightly worse. Both results are, however, within the range between 169—174. This means that both results are almost regarded as satisfactory.

4 PRACTICAL EXPERIMENT

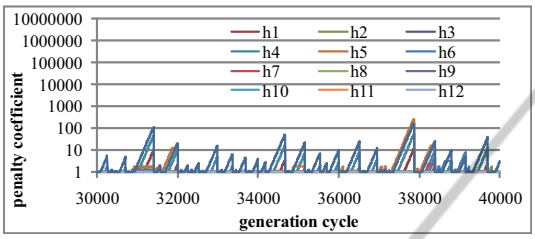
We have tried computational experiment of the nurse scheduling with practical data. In order to compare exactly, we have tried to optimize the ten times under each condition. Figures 5 (a)—(c) show examples

5 CONCLUSIONS

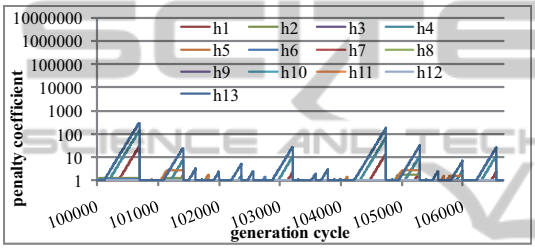
This paper has shown a technique of nurse scheduling by using CGA. We have discussed the case that the nurse schedule has been changed in the past weeks. To reoptimize the changed schedule, we have de-



(a) from 0-th generation to 10000-th generation



(b) from 30000-th generation to 40000-th generation



(c) from 100000-th generation to the final generation

Figure 5: The variation of the penalty coefficients. Each vertical axis shows on a logarithmic scale.

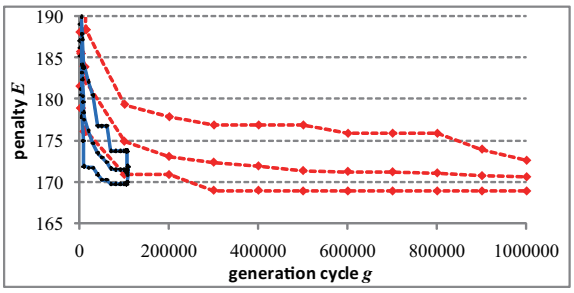


Figure 6: Comparison of the optimization progress between CGA with the periodic mutation and CGA with PA.

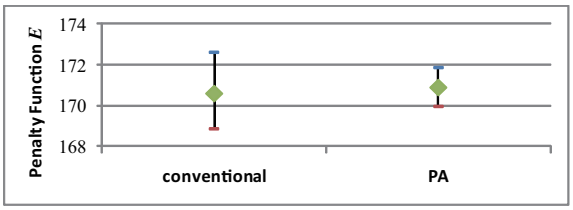


Figure 7: Comparison of the optimization result between the periodic mutation and PA.

fined a penalty function performing the difference between the original schedule and the optimizing schedule. Therefore we need new techniques to search for good schedule effectively. We have proposed a technique adjusting the penalty coefficient depending on the optimization progress, PA. This technique is implemented with the mutation depending on the optimization speed. By means of PA, the optimization finishes within one-tenth generation cycles by the conventional periodic mutation technique.Thus, the effectiveness of PA is confirmed.

ACKNOWLEDGEMENTS

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

REFERENCES

D. E. Goldberg (1989). *Genetic Algorithm in Search, Optimization and Machine Learning.* New York.

E. K. Burke, P.De Causmaecker, S. G. (2001). Fitness evaluation for nurse scheduling problems. In *Proceedings of the 2001 Congress on Evolutionary Computation*.

M. Ohki (2010a). Effective mutation operator for nurse scheduling by cooperative ga and its parallel processing. In *19th Int. ACM Workshop on Parallel Architectures and Bioinspired Algorithms*, pages 1–8.

M. Ohki, S.Uneme, H. (2010b). Effective mutation operator and parallel processing for nurse scheduling. In *Studies in Computational Intelligence*, volume 299, pages 229–242. DOI: 10.1007/978-3-642-13428-9_10.

M. Ohki, A. Morimoto, K. (2006). Nurse scheduling by using cooperative ga with efficient mutation and mountain-climbing operators. In *3rd Int. IEEE Conference Intelligent Systems, pp.164-169*.

M. Ohki, S. Uneme, S. M. (2007). Effective genetic operators of cooperative genetic algorithm for nurse scheduling. In *4th Int. INSTICC Conference on Informatics in Control, Automation and Robotics*, pages 347–350.

S. Uneme, H. Kawano, M. (2008). Nurse scheduling by cooperative ga with variable mutation operator. In *Proc. of 10th ICEIS, INSTICC*, pages 249–252.

T. Itoga, N. Taniguchi, Y. K. (2003). An improvement on search efficiency of cooperative ga and application on nurse scheduling problem. In *Proc. of 12th Intelligent System Symposium*, pages 146–149.