EFFECTIVE GENETIC OPERATORS OF COOPERATIVE GENETIC ALGORITHM FOR NURSE SCHEDULING

Makoto Ohki, Shin-ya Uneme, Shigeto Hayashi and Masaaki Ohkita
Department of Electric and Electronic Engineering, Faculty of Engineering, Tottori University

Keywords: Genetic Algorithm, Cooperative GA, Mutation Operator, Virus Operator, Nurse Scheduling, Scheduling Problem.

Abstract: This paper proposes effective genetic operators for cooperative genetic algorithm (GA) to solve a nurse scheduling problem. A clinical director of a medical department makes a duty schedule of all nurses of the department every month. Such the scheduling is very complex task. It takes one or two weeks to create the nurse schedule even by a veteran director. In conventional ways using the cooperative GA, a crossover operator is only employed for the optimization, because it does not lose consistency between chromosomes. We propose a mutation operator and a virus operator for the cooperative GA, which does not lose consistency of the nurse schedule. The cooperative GA with these new operators has brought a surprisingly good result, it has never been brought by the conventional algorithm.

1 INTRODUCTION

In hospitals, 15-30 nurses are working in a medical section such as the internal medicine department or the pediatrics department. A clinical director of the department constitutes a schedule of all the nurses every month. Constituting the nurse schedule (Goto, 1993, Ikegami 2001), or the nurse scheduling, is very complex task. In our investigation, even a veteran director needs one or two weeks for the nurse scheduling. To reduce such the problem, a computer software creating the nurse schedule is developed, and it is recentlly used at many hospitals. Those software must find the better schedule in the shorter time.

In conventional ways (Goto, 1993, Kawanaka 2002, Inoue, 2002, Itoga, 2003) using the cooperative genetic algorithm (CGA), a crossover operator is only employed for the optimization, because it does not lose consistency between chromosomes. Theirfore, the conventional technique optimizes the nurse schedule at slow speed, and it is difficult to acquire a good solution.

We propose mutation and virus operators for the CGA, which do not lose consistency of the nurse schedule. The CGA with these new operators has brought a surprisingly good result, it has never been brought by the conventional algorithm.

2 NURSE SCHEDULING BY CGA WITH CROSSOVER

In the nurse scheduling, we have to consider many requirements, for example, duty load of each nurse, fairness of assignment of day time and night duty, intensiveness of night duty and so on. These requirements are implemented by twelve penalty functions in this paper. These twelve penalty functions, F_1-F_{12}, are classified into four partial penalty groups, G_1-G_4. Finally, we define an total penalty function, E, and the partial penalty functions, G_1-G_4, as follows,

$$E(\alpha) = \sum_k G_k^{\ 2}, \tag{1}$$

$$G_1 = \sum_i \left(h_{11}F_{1i} + h_{12}F_{4i} + h_{13}F_{5i}\right), \tag{2}$$

$$G_2 = \sum_i \left(h_{21}F_{2i} + h_{22}F_{3i} + h_{23}F_{6i}\right), \tag{3}$$

$$G_3 = \sum_j \left(h_{31}F_{7j} + h_{32}F_{8j} + h_{33}F_{9j}\right), \tag{4}$$

and

$$G_4 = \sum_j \left(h_{41}F_{10j} + h_{42}F_{11j} + h_{43}F_{12j}\right), \tag{5}$$

where h_{pq} (p=1,2,3 or 4, q=1,2 or 3) denotes energy coefficients as follows,

h_{11}=0.2, h_{12}=0.4, h_{13}=0.4, h_{21}=0.2, h_{22}=0.2, h_{23}=0.4,

Ohki M., Uneme S., Hayashi S. and Ohkita M. (2007).
EFFECTIVE GENETIC OPERATORS OF COOPERATIVE GENETIC ALGORITHM FOR NURSE SCHEDULING.
In *Proceedings of the Fourth International Conference on Informatics in Control, Automation and Robotics*, pages 347-350

h_{31}=0.1, h_{32}=0.1, h_{33}=0.1, h_{41}=0.7, h_{42}=0.7, h_{43}=0.7. Each nurse schedule of one month is evaluated by the following penalty function,

$$H_i = h_{11}F_{1i} + h_{12}F_{4i} + h_{13}F_{5i} + h_{21}F_{2i} + h_{22}F_{3i} + h_{23}F_{6i} \quad (6)$$

In the cooperative GA, the population represents the whole nurse schedule of one month. Each individual is coded in a chromosome which shows one-month schedule of one nurse. The individual consists of the sequence of the duty symbols as shown in Fig.1. The duty sequence consists of thirty fields, since one month includes thirty days in this practical example. Each individual expresses one-month schedule of the nurse n_i. There are not two or more individuals including the same nurse's schedule in the population. An example of the population is shown in Fig.2.

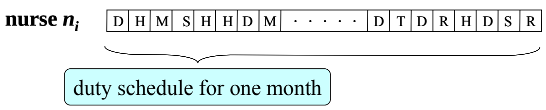

Figure 1: An individual coded in chromosome giving duty schedule of one nurse for one month.

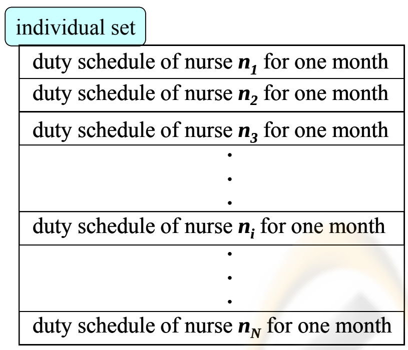

Figure 2: Population including one-month schedules of each nurse.

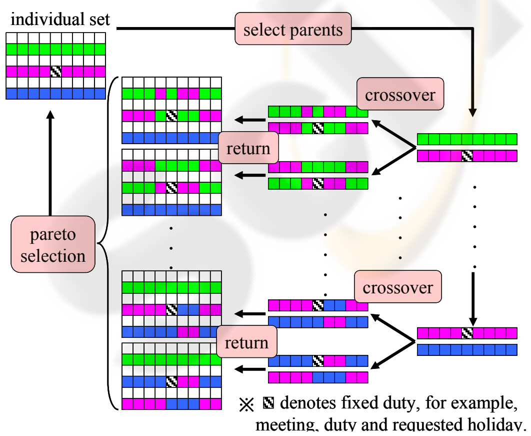

Figure 3: Optimization cycle of the CGA with the crossover operator.

Initially, the population is randomly generated as satisfying the necessary number of nurses on every duty at every date. In this paper, the necessary number of nurses is specified as ten, six and five for day time duty on a week day, Saturday and Sunday respectively, three for a semi night duty and three for night duty respectively. An overview of the crossover operator is shown by Fig.3. First, two hundreds pairs of the individual are selected as a parent for the crossover. One of the pair is selected from the population under the roulette selection manner. The roulette selection gives an individual with higher energy, H_i. This manner tends to select the worse individual. Another one of the pair is randomly selected from the population. Exchanging parts of the individuals divided at two crossover points, two new pairs of individual are generated as children. This exchanging process does not exchange the fixed duties, such as the meeting, the training and the requested holiday. These child pairs are temporally returned to the original nurse position respectively. The temporal population with the children is performed by the energy function, E. After all the pairs of individual have been performed, one pair giving the smallest energy is selected for the next generation.

3 NEW GENETIC OPERATORS

In this paper, two new genetic operators are proposed to accelerate the nurse scheduling by CGA. One of them is a mutation operator. In the conventional way, it is considered that the mutation loses the consistency of the population and does not work effectively. However, the mutation is a very strong genetic operator to widely search in the solution space. In this paper, new effective mutation operator is proposed for the cooperative GA which does not lose the consistency of the schedule. The aim of mutation is to bring small change into the population. However if one of the duty fields is randomly changed, the whole schedule become meaningless thing, which is very hard to recover by a genetic operation. Therefore, the mutation operator must preserve the number of nurses in every duty at all date. The basic operation of the mutation is shown in Fig.4. One of dates is randomly selected. Two nurses at the same date are decided. Finally, the duties of these two positions are exchanged. If one of the selected duties is the fixed duty, another nurse is randomly selected again. The basic mutation operation is applied to CGA as shown in Fig.5.

We also propose a virus operator as another new technique. When a virus in immunology is infected within a cell, it overwrites in a part of gene forcibly. The virus copies own gene pattern in a genetic part of the cell in many cases. Our virus operator simulates this work. An aim of the virus operator is to replace some individuals with a good thing forcibly when the optimization is stagnant.

In normal optimization cycle, the CGA searches by using the crossover and the mutation operators and preserves the best performing individuals after the crossover. When the mutation operator is executed G_V times, the virus operator is applied instead of the mutation operator. One of individual who gives a bad penalty is selected by using a roulette selection manner. The virus operator overwrites the best performed individual onto the selected individual as shown in Fig.6.

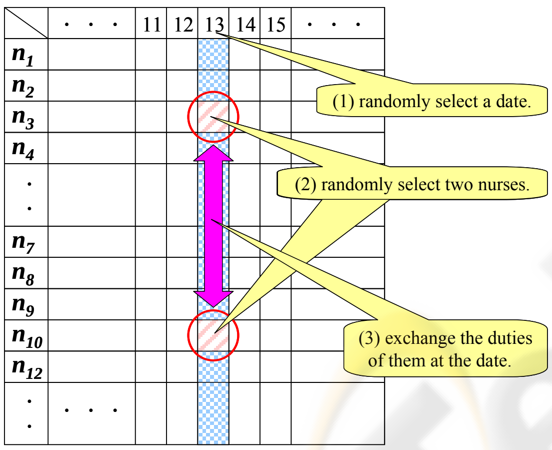

Figure 4: Primitive operation of the mutation operator.

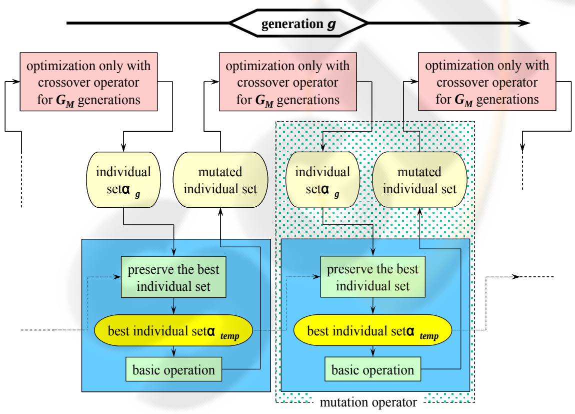

Figure 5: Mutation operator executed by a period for G_M generations.

4 COMPARISON OF GENETIC OPERATORS

We have tried three types of the CGA in this paper, the CGA only with the crossover operator, the CGA with the crossover and the mutation operators and the CGA with those three operators. An optimization of each algorithm is performed for One hundred thousands generations. It takes about ten minutes for one optimization. Ten times of trials are carried out under each condition.

Firstly, a difference of the mutation period is examined. As shown by Fig.7, the mutation period, G_M, is tried to change from fifty to two thousands generations. We have decided that the mutation period should be set at 150. Average value of the energy function for ten times trials is shown in Fig.8. The mutation with a period at less than 200 generation effectively works.

We have examined the virus operator. Different viral infection frequencies are examined as shown by Fig.9. The viral infection frequency is able to be set in a considerably wide range.

We have investigated the optimization with the virus operator in detail. After twenty or thirty thousands generations, penalty function, F_1, has big value, and the value of all other penalty functions is comparatively small. Then we have modified a part of the virus operator: when the best individual is preserved after the crossover operation, the following penalty function is applied,

$$H_i = h_{12}F_{4i} + h_{13}F_{5i} + h_{21}F_{2i} + h_{22}F_{3i} + h_{23}F_{6i} \,. \quad (7)$$

As shown by Fig.10, the modified virus operator gives good results with any virus frequency.

The modified virus operator does not effectively work after thirty thousands generation as shown in

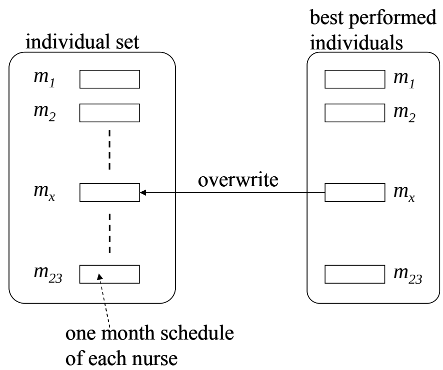

Figure 6: The virus operator forcibly overwrites an one-month schedule of one of nurses selected in the roulette selection manner, when the optimization is stagnant.

Fig.11. In contrast, the mutation operator slowly searches. Then we apply the modified virus operator until thirty thousands generations. After that, the mutation operator is only applied to the CGA with the crossover. As shown in Fig.11, the rearranged technique gives the best result.

5 CONCLUSION

We have proposed two new genetic operators for the CGA applied to the nurse scheduling. The only one search technique of the conventional CGA is the crossover operator, because it dose not lose the consistency of the nurse schedule. In contrast, the new techniques which we proposed in this paper give good results without losing the consistency. By means of these new techniques, the optimization of the nurse schedule has been accelerated.

REFERENCES

Goto, T., Aze, H., Yamagishi, M., Hirota, M. and Fujii, S. 1993, *Application of GA, Neural Network and AI to Planning Problems*, NHK Technical report, No.144.

Ikegami, A. 2001, *Algorithms for Nurse Scheduling*, Proc. of 11th Intelligent System Symposium.

Kawanaka, S., Yamamoto Y., Yoshikawa D., Shinogi T. and Tsuruoka N., 2002, *Automatic Generation of Nurse Scheduling Table Using Genetic Algorithm*, Trans. on IEE Japan, Vol.122-C.

Inoue, T., Furuhashi, T., Maeda H. and Takabane, M., 2002, *A Study on Interactive Nurse Scheduling Support System Using Bacterial Evolutionary Algorithm Enegine*, Trans. on IEE Japan, Vol.122-C.

Itoga, T., Taniguchi N., Hoshino Y. and Kamei K., 2003, *An Improvement on Search Efficiency of Cooperative GA and Application on Nurse Scheduling Problem*, Proc. of 12th Intelligent System Symposium.

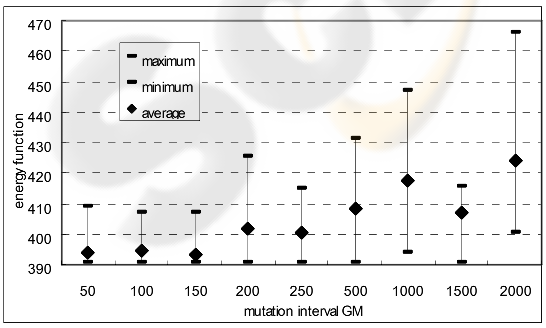

Figure 7: Comparison of the mutation period.

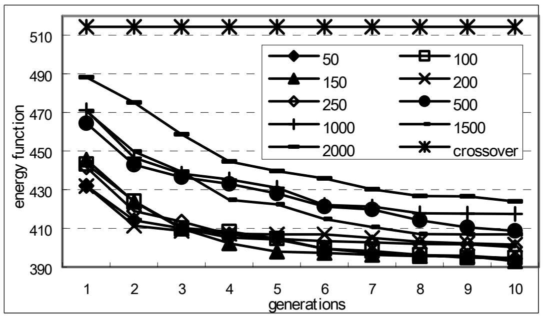

Figure 8: Average value of the energy function for ten times trials. The unit of the horizontal line is ten thousand.

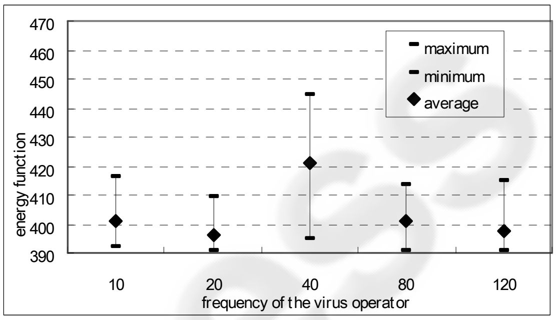

Figure 9: Comparison of the frequency of the virus operator.

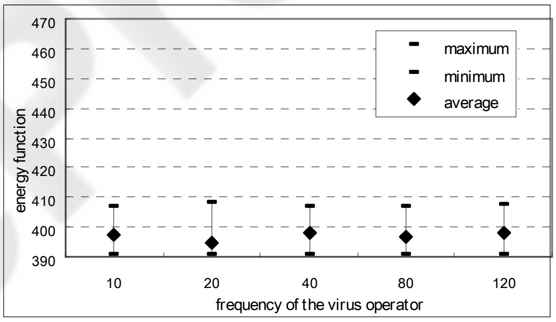

Figure 10: Comparison of the frequency of the modified virus operator.

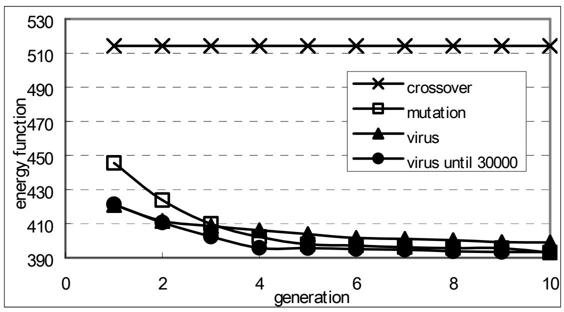

Figure 11: Comparison of the optimization process among CGA only with the crossover operator, CGA with the mutation operator, CGA with the modified virus operator and CGA applied with the modified virus operator until 30000 generations. The unit of the horizontal line is ten thousand.