International Journal Data Modelling and Knowledge Management
Vol. 6 No. 1 (January-June, 2021)

Submitted: 12th October 2020 *Revised: 14th December 2020* *Accepted: 30th December 2020*

An Effective Virus Operator for Cooperative GA applied to Nurse Scheduling Problem

Makoto Ohki*
4, 101 Koyama-Minami, Tottori, Tottori 680-8552 Japan

Authors emails
mohki@ele.tottori-u.ac.jp
Fax.:+ 81 857 31 5231

ABSTRACT

In this paper, we propose effective genetic operators so that the cooperative genetic algorithm (GA) can solve the nurse scheduling problem. A clinical director of a medical department must make a duty schedule for all nurses in the department every month. This scheduling task is very complex. It takes one or two weeks to create the nurse schedule even for a veteran director. In the conventional usage of cooperative GA, a crossover operator is employed only for optimization to retain consistency between chromosomes. We propose a virus operator for the cooperative GA that ensures the consistency of the nurse schedule. The cooperative GA with the new operator yields surprisingly good results that are never suggested by the conventional algorithm.

1. INTRODUCTION

General hospitals consist of several sections such as the internal medicine department and the pediatrics department. About fifty to thirty nursing staff are assigned to each section. The section manager develops a roster, or a shift schedule, of all nurses of her/his section every month. Our interviews of the staff of several hospitals found that the manager considers more than fifteen requirements in creating the schedule. Scheduling nurses is, therefore, a very complex task. We call this problem the Nurse Scheduling Problem (NSP). In the interview, even a veteran manager has to spend one or two weeks to complete the nurse scheduling. This represents a great loss of time and effort. Therefore, general hospitals are starting to demand computer software that can solve NSP (Goto, 1993; Berrada, 1996; Takaba, 1998; Ikegami, 2001; Burke, 2001a; Kawanaka, 2002; Inoue, 2002; Itoga, 2003; Cheang, 2003; Burke, 2004a; Ernst, 2004; Burke, 2004b; Li, 2004; Bard, 2005; Oezcan, 2005; Burke, 2006; Bard, 2007).

In an early study (Goto, 1993), the nurse scheduling problem, defined as a discrete planning problem, is solved by using the Hopfield-model type-neural network. Berrada *et al.* (Berrada, 1996) 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. (Takaba, 1998) provides a simple editing tool and simple GA for the nurse scheduling under Visual Basic environment. There are several techniques (Ikegami, 2001; Inoue, 2002; Bard, 2005; Bard, 2007) that require the user to modify or select the nurse schedule in the middle or the final stage of the optimization. Although these researches are theoretical trial, these are not effective in a practical sense. Burke *et al.* apply a memetic approach to the nurse scheduling problem (Burke, 2001a; Burke, 2004a; Burke, 2004; Burke, 2006). Burke *et al.* (Burke, 2001a) also define a technique to evaluate the nurse schedule. Croce *et al.* (Croce, 2010) proposes a variable neighborhood search technique for the nurse scheduling. However, the scheduling problem defined in this manuscript is too easy. And the technique proposed in this manuscript is applied to a private hospital in Italy. Real problem of the nurse scheduling in the general hospital is not so easy and very hard to solve. Some of these techniques are implemented as a 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 (Ohki, 2006; Ohki, 2007; Uneme, 2008; Ohki, 2012).

In the real world, there are cases that nurses attend on a different day from the original schedule because of the circumstances of another 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 have discussed such a case that the nurse schedule has been changed in the past weeks of the current month (Ohki, 2006; Ohki, 2007; Uneme, 2008; Ohki, 2012). 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 acquire 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 that use commercial software for nurse scheduling. Managers are dissatisfied with the shift schedule generated by 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) (Itoga, 2003). CGA is a kind of Genetic Algorithm (GA) (Goldberg, 1989), and powerful optimizing algorithm for such a combinatorial optimization problem. In the normal GA, individuals compete each other and superior individuals are preserved. On the other hand, the individuals cooperate each other and the optimization of whole population progresses in CGA.

We have proposed effective mutation operators for CGA to be applied to NSP (Ohki, 2012). 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 have proposed an effective mutation operator keeping the consistency for the CGA (Uneme, 2008). 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 have proposed a simple mutation operator activated periodically. We call this operator the periodic mutation operator.

In this paper, we propose a virus operator for the cooperative GA, which does not lose consistency of the nurse schedule. The cooperative GA with the virus operator has brought a surprisingly good result, it has never been brought by the conventional algorithm.

2. EVALUATION OF THE NURSE SCHEDULE

We have interviewed to several real general hospitals. By means of the interviews, a method to evaluate the nurse schedule is clarified as follows (Ohki, 2006; Ohki, 2007; Uneme, 2008; Ohki, 2012). 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. The detail of these penalty functions are explained in the manuscript (Ohki, 2012).

To evaluate the work load of each nurse, we define a penalty function F_{1i} for three consecutive days of shift content. It is not preferable for the night shifts to be assigned to some nurse intensively. To suppress this undesirable situation, we define a penalty function F_{2i} to prohibit the X night shift or more for the consecutive Y days. 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, a penalty f_{3ij} is assigned to 1. We define a penalty function, F_{3i}, equal to the sum total of f_{3ij} from $j = 1$ to $j = D$, to implement such the prohibition, where D denotes the number of total shift days.

The number of the shifts should be impartially assigned among nurses. A total nursing level falls, if many shifts are concentrated to particular nurses. We define penalty functions F_{4i} and F_{5i} to suppress unevenness of the number of shifts among nurses. The functions F_{4i} and F_{5i} are concerning the numbers of holidays and the number of night shifts respectively. If the shifts are assigned to particular nurses on many consecutive days, total nursing level falls. We define a penalty function F_{6i} to restrain assignment of the shift on many consecutive shift days.

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 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 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. To restrain such the unfavorable affinity, we define a 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, the nursing level at the midnight shift falls remarkably. To restrain such the unfavorable situation, we define a penalty function, F_{11j}. 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}.

At the real hospital, the shift schedule which optimized before the beginning of the current month is often changed day by day. Such changes of the schedule lead to the disproportion of the number of the shift days. It causes the overwork of particular nurses, if such unexpected situation is ignored. This undesirable situation yields the fall of the nursing level, but may lead to medical accidents as well. To restrain such an undesirable situation, we consider on the reoptimization of the shift schedule of the remainder of the current month. First, we assume that we have had the well 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. With considering the circumstances of the nurses, the shift schedule should not be changed as much as possible. 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. 1. In this figure, the triangles show the changes of the schedule.

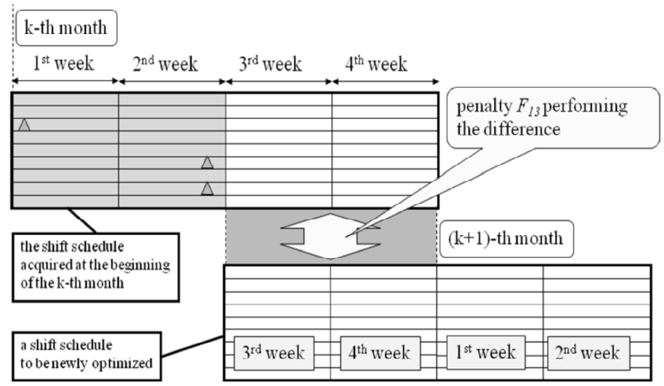

Figure 1: We expand NSP to accept some changes in the past two weeks. This figure shows an example when the two weeks have past, the coming four weeks are optimized to restrain inconvenience because of the changes.

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

$$E(g) = \sum_{i=1}^{M}\sum_{k=1}^{6} h_k F_{k\,i} + \sum_{j=1}^{D}\sum_{k=7}^{12} h_k F_{k\,j} + h_{13}F_{13} \qquad (1)$$

where h_1 --- h_{13} denote penalty coefficients.

3. NURSE SCHEDULING BY CGA

3.1. Coding of the Nurse Schedule

In the nurse scheduling by CGA, an individual and its group, or the population, are defined shown in Fig. 2. 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 do 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 and h denote a training shift, a meeting and a requested holiday accepted by the chief nurse, where these are treated as a daytime shift term. Symbol, R, is a requested holiday which confirmed by the manager.

3.2. Basic Algorithm to Optimize the Nurse Schedule by Using CGA

The basic algorithm of the CGA is as shown in Fig. 3 (Ohki, 2006; Ohki, 2007; Uneme 2008). CGA applies the crossover operator to the population and searches so that a penalty of the whole population becomes

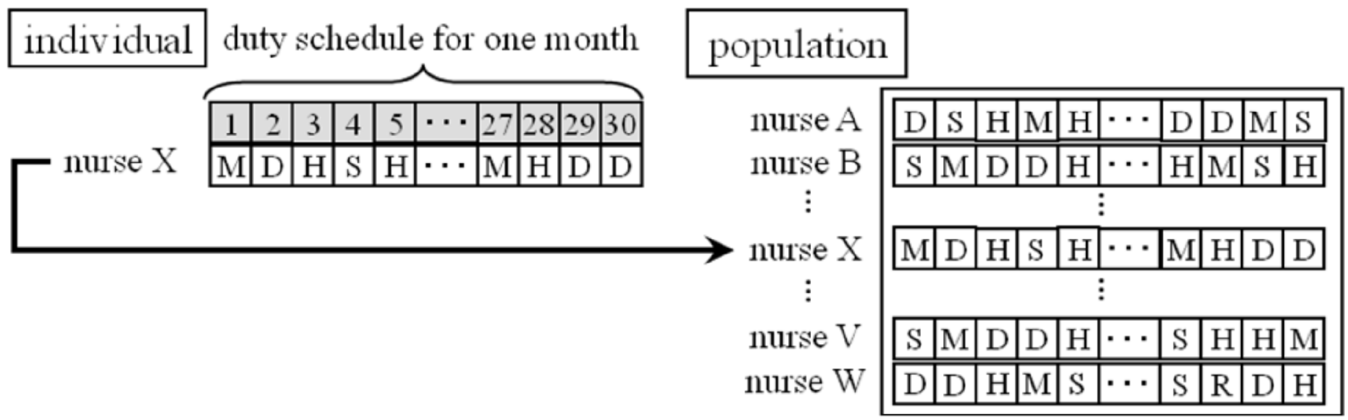

Figure 2: Definition of chromosome and population. The X-th individual coded into chromosome denotes one month shift schedule of the X-th nurse. The population including one month schedule of the current nurse organization.

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 $E(g)$. 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.

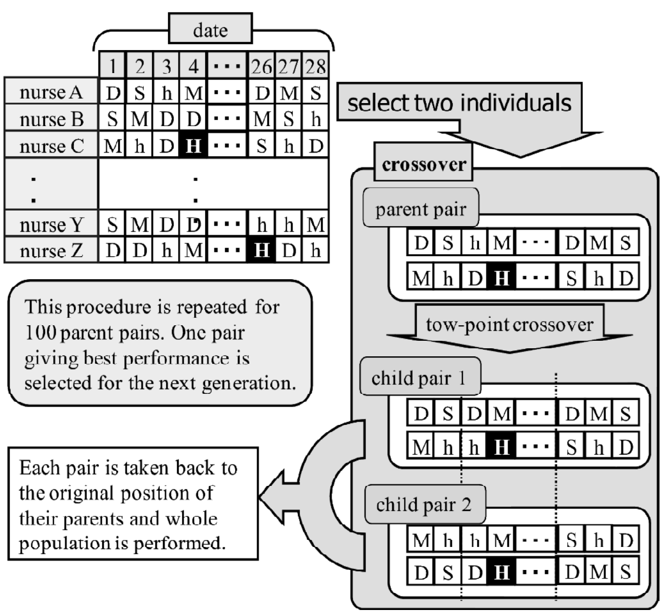

Figure 3: One generation cycle of the CGA to optimize by using a crossover operator.

3.3. Periodic Mutation Operator

The mutation operator is periodically activated every G_M generation cycles. Fig. 4 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. We have tried computational experiments under a condition that 30 nurses belong the section. This experiment has been carried out under harder condition about holiday acquisition and nurse affinity. Fig. 5 shows the maximum, the average and the minimum value of the total penalty function given by the CGA with the periodic mutation. The mutation period is effective on narrow range from 50 to 250.

The computing time is recorded in ten trials under the condition that the mutation period is defined as 150. The minimum, the averaged and the maximum computing time is 8239 sec, 8356 sec and 9134 sec respectively. The computing time in the case when the periodic mutation is applied is almost same to when the mutation depending on the optimization speed is applied.

4. VIRUS OPERATOR

We propose a virus operator as a new technique. When a virus in immunology is infected within a cell, it

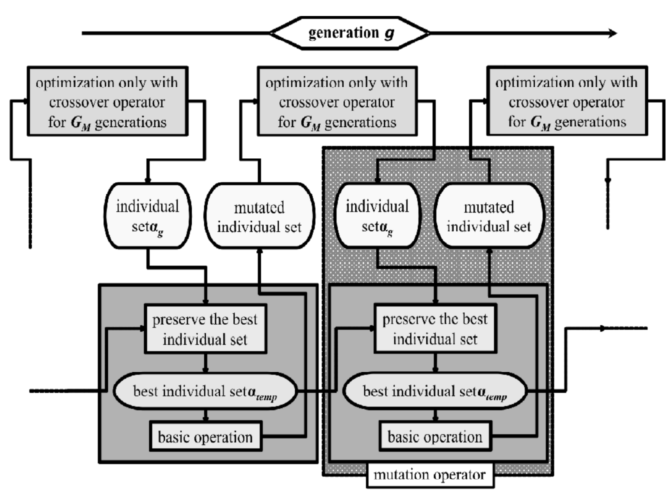

Figure 4: Process flow of the periodic mutation operator.

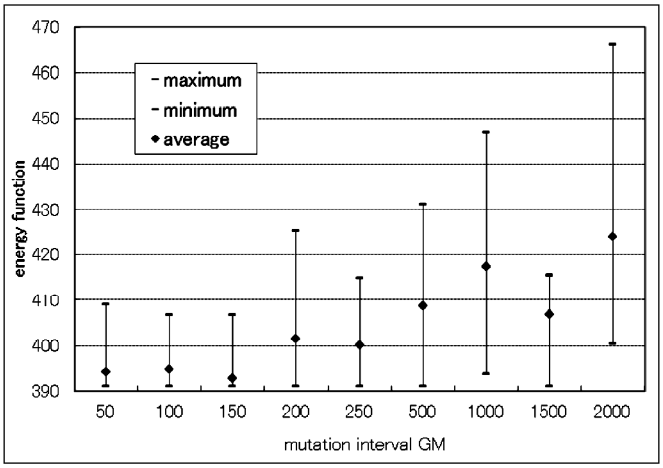

Figure 5: Optimization results by the periodic mutation operator with several mutation periods. We have tried to set the mutation period from 50 to 5000

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.

The overview of an operation of the virus operator is as shown in Fig. 6. 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.

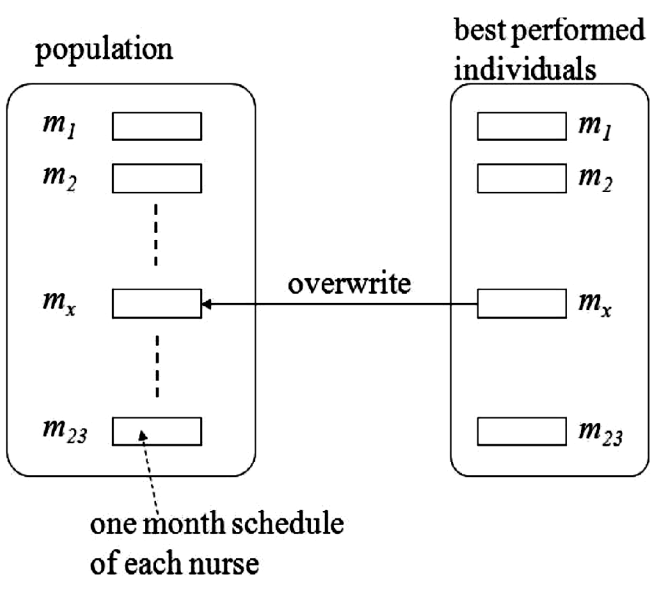

Figure 6: An overview of the operation of the virus operator.

We have tried two types of the CGA in this paper; 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 thousand generations. It takes about ten minutes for one optimization. Ten times of trials are carried out under each condition.

We have examined the virus operator. Different viral infection frequencies are examined as shown by Fig. 7. The viral infection frequency is able to be set in a considerably wide range.

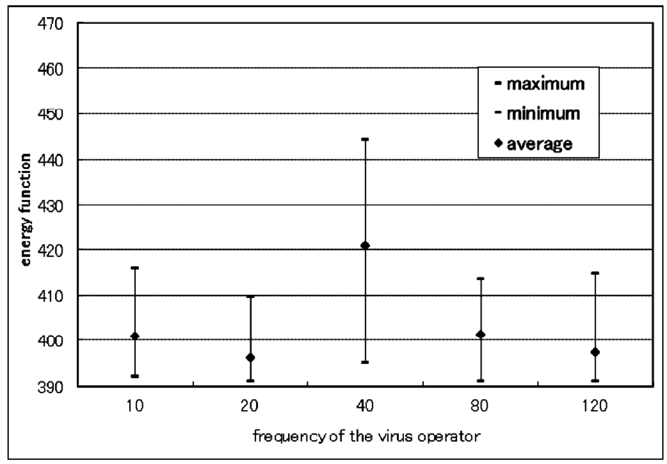

Figure 7: Comparison of the frequency of the first virus operator.

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

$$H_i = \sum_{k=2}^{6} h_i F_{k\ i} \tag{2}$$

As shown by Fig. 8, the modified virus operator gives good results with any virus frequency.

The modified virus operator does not effectively work after thirty thousand generation as shown in Fig. 9. In contrast, the mutation operator slowly searches. Then we apply the modified virus operator until thirty thousand generations. After that, the mutation operator is only applied to the CGA with the crossover. We call this technique a hybrid technique. As shown in Fig. 9, the hybrid technique gives the best result. The maximum, the average and the minimum values of the total penalty function after the final generation are 398.26, 394.00 and 390.92. This minimum value is similar to the value shown in Fig. 4 (G_M = 150). The maximum and the average value are both better than

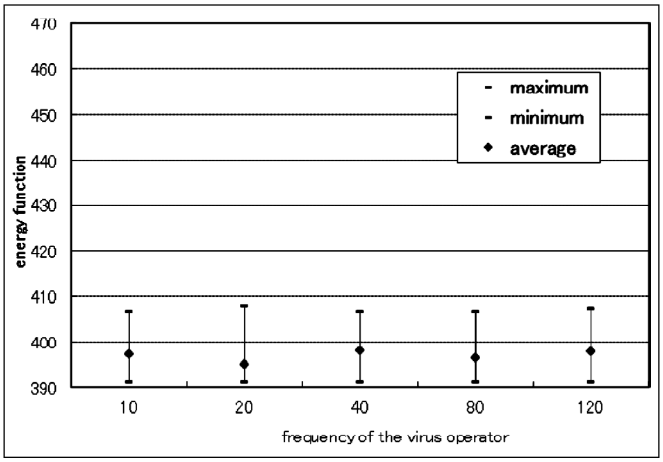

Figure 8: Comparison of the frequency of the modified virus operator.

the values shown in Fig. 4 ($G_M = 150$). The hybrid technique makes the optimization of the schedule converge to good solution faster.

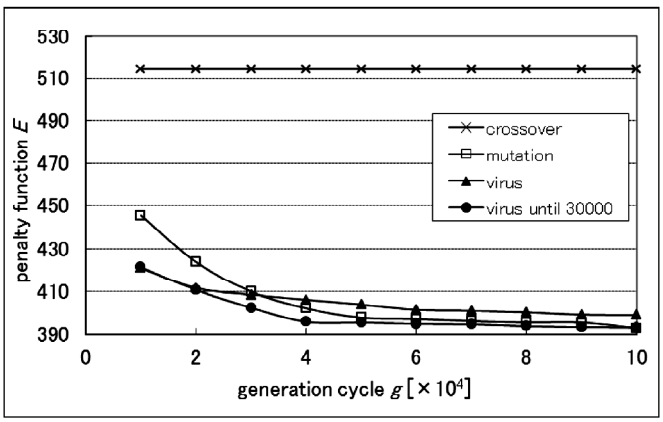

Figure 9: Comparison of the optimization process among CGA only with the crossover operator, CGA with the mutation operator, CGA with the modified virus operator and CGA applied with the modified virus operator until 30000-th generation.

5. CONCLUSION

This paper has introduced a nurse scheduling method that is based on CGA. We have proposed the virus operator for the CGA applied to the nurse scheduling. The only one search technique of the conventional CGA is the crossover operator, because it do not lose the consistency of the nurse schedule. In contrast, the virus operator which we proposed in this paper give good results without losing the consistency. However, the virus operator seems to be inferior to the periodic mutation operator. Then we have proposed the modified virus operator. This new method gives stable results than periodic mutation operator. Finally, we have proposed the hybrid technique. In the hybrid technique, CGA searches the schedule with the modified virus operator until the 3000-th generation cycle, and then, CGA searches the schedule with the periodic mutation operator. By means of the hybrid technique, the optimization of the nurse schedule has been accelerated.

In the future, the exchanging generation, 3000-th generation cycle in this paper, should be automatically decided. And also, 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 executed in parallel. This is an aspect of the macro parallelization.

ACKNOWLEDGEMENT

This research work has been supported by Tottori University Electronic Display Research Center (TEDREC). I would like to thank to cats Civita, Blacky, Blanc and Caramel gave his the healing daily life.

REFERENCES

[1] Goldberg, D.E. (1989), "Genetic Algorithm in Search, Optimization and Machine Learning", *Addison-Wesley, New York.*

[2] Goto, T., Aze, H., Yamagishi, M., Hirota, M. & Fujii, S. (1993), "Application of GA, Neural Network and AI to Planning Problems", *NHK Technical Report*, 144, 78-85.

[3] Berrada, I., Ferland, J.A. & Michelon, P. (1996), "A Multi-objective Approach to Nurse Scheduling with both Hard and Soft Constraints", *Socio-Econ. Plann. Sci*: *30*, 3, 183-193.

[4] Takaba, M., Maeda & H., Sakaba, N. (1998), "Development of a Nurse Scheduling System by a Genetic Algothm", *Proc. of 18th JCMI.*

[5] Ikegami, A. (2001), "Algorithms for Nurse Scheduling", *Proc. of 11th Intelligent System Symposium*, 477-480.

[6] Burke, E. K. & Cowling, P. (2001a), "A Memetic Approach to the Nurse Rostering Problem", *Applied Intelligence, 15*, 199-214.

[7] Burke, E.K., De Causmaecker, P., Petrovic, S. & Berghe, G.V. (2001b), "Fitness Evaluation for Nurse Scheduling Problems", *Proc. of the 2001 Congress on Evolutionary Computation.*

[8] Kawanaka, S., Yamamoto Y., Yoshikawa D., Shinogi T. & Tsuruoka N. (2002), "Automatic Generation of Nurse Scheduling Table Using Genetic Algorithm", *Trans. on IEE Japan*: 122-C, 6, 1023-1032.

[9] Inoue, T., Furuhashi, T., Maeda H. & Takabane, M. (2002), "A Study on Interactive Nurse Scheduling

Support System Using Bacterial Evolutionary Algorithm Engine", *Trans. on IEE Japan*: 122-C, 10, 1803-1811.

[10] Itoga, T., Taniguchi N., Hoshino Y. & Kamei K. (2003), "An Improvement on Search Efficiency of Cooperative GA and Application on Nurse Scheduling Problem", *Proc. of 12th Intelligent System Symposium*, 146-149.

[11] Cheang, B., Li, H., Lim, A. & Rodrigues, B. (2003), "Nurse Rostering Problems - a bibliographic survey", *Europiean Journal of Operational Research, 151*, 447-460.

[12] Burke, E.K., De Causmaecker P., Berghe, G.V. & Lnadeghem, H. (2004a), "The State of the Art of Nurse Rostering", *Journal of Scheduling, 7*, 441-499.

[13] Ernst, A.T., Jiang, H., Krishnamoorthy, M., Owens, B. & Sier, D. (2004), "An Annotated Bibliography of Personnel Scheduling and Rostering", *Annals of Operations Research, 127*, 21-144.

[14] Burke, E.K., De Causmaecker, P. & Berge, G.V. (2004b), "Novel Meta-Heuristic Approaches to Nurse Rostering Problems in Belgian Hospitals", *J. Leung Handbook of Scheduling Algorithms, Models and Performance Analysis.*

[15] Li, J., Aickelin U. (2004), "The Application of Bayesian Optimization and Classifier Systems in Nurse Scheduling", *Proc. of the 8th International Conference on Parallel Problem Solving from Nature,* LNCS 3242, 581-590.

[16] Bard, J.F. & Purnomo, H.W. (2005), "Preference Scheduling for Nurses using Column Generation", *European Journal of Operational Research, 164*, 510-534.

[17] Oezcan, E. (2005), "Memetic Algorithm for Nurse Rostering", *Computer and Information Sciences - ISCIS 2005 Lecture Notes in Computer Science*, 37332005, 482-492.

[18] Ohki, M., Morimoto, A. & Miyake, K. (2006), "Nurse Scheduling by Using Cooperative GA with Efficient Mutation and Mountain-Climbing Operators", *3rd Int. IEEE Conference Intelligent Systems*, 164-169.

[19] Burke, E.K., De Causmaecker, P., Petrovic, S. & Berge, G.V. (2006), "Metaheuristics for Handling Time Interval Coverage Constraints in Nurse Scheduling", *Applied Artificial Intelligence, 20*, 3.

[20] Ohki, M., Uneme, S., Hayashi, S. & Ohkita, M. (2007), "Effective Genetic Operators of Cooperative Genetic Algorithm for Nurse Scheduling", *4th Int. INSTICC Conference on Informatics in Control, Automation and Robotics*, 347-350.

[21] Bard, J.F. & Purnomo, H.W. (2007), "Cyclic Preference Scheduling of Nurses Using a Lagrangian-Based Heuristic", *Journal of Scheduling, 10*, 5-23.

[22] Uneme, S., Kawano, H. & Ohki, M. (2008), "Nurse Scheduling by Cooperative GA with Variable Mutation Operator", *Proc. of 10th ICEIS, INSTICC*, 249-252.

[23] Croce, F.D. & Salassa, F. (2010), "A Variable Neighborhood Search based Mataheuristic for Nurse Rostering Problems," *Proc. of 8th International Conference on the Proctice and Theory of Automated Timetabling,* 167-175.

[24] Ohki, M. (2012), "Periodic Mutation Operator for Nurse Scheduling by Using Cooperative GA," *International Journal of Applied Evolutionary Computation, 3*, 3, 1-16, http://www.igi-global.com/article/periodic-mutation-operator-nurse-scheduling/68830.