ISSN: 1803-3814

SHIFT SCHEDULING OF SHORT TIME WORKERS IN LARGE-SCALE HOME IMPROVEMENT CENTER BY USING COOPERATIVE EVOLUTION

Makoto Ohki

Department of Information and Electronics, Graduate School of Tottori University,
4, 101 Koyama-Minami, Tottori, Tottori 680-8552 Japan
mohki@ele.tottori-u.ac.jp

Abstract: *There are a lot of large-scale Home Improvement Center (HIC) in Japan. In the large-scale HIC, about hundred short time workers are registered. And about forty workers are working every day. A manager creates a monthly shift schedule. The manager selects the workers required for each working day, assigns the working time and break time for each worker and also work placement. Because there are many requirements for the scheduling, the scheduling consumes time costs and efforts. Therefore, we propose the technique to create and optimize the schedule of the short time workers in order to reduce the burden of the manager. A cooperative evolution is applied for generating and optimizing the shift schedule of short time worker. Several problems has been found in carrying out this research. This paper proposes techniques to automatically create and optimize the shift schedule of workers in large-scale HIC by using a Cooperative Evolution (CE), to solve the situation that many workers concentrate in a specific time zone, and to solve the situation where many breaks are concentrated in a specific break time zone, and an effective mutation operators.*

Keywords: *Scheduling, Large-Scale Home Improvement Center, Cooperative Evolution, Crossover Operator, Worker Number Adjustment, Break Time Adjustment, Mutation Operator.*

1 Introduction

A Home Improvement Center (HIC) in Japan usually deals with a variety of products such as a daily necessities, groceries, pet supplies, power tools, tool materials, medicins and gardening materials. There are a lot of large-scale HIC in Japan. About 50-100 short time workers are registered and working in the large-scale HIC. The main jobs of the worker are cashiering (checker), packing bags (sacker), a services at the service counter (SC), dealing with horticultural products (GR), dealing with material goods (MR) and so on. A manager of the HIC creates a monthly shift schedule based on desired working hours sent from the worker and expertise of workers with considering various conditions such as day of the week, weather condition, the season, necessary number of personnel, personel budget. It is very difficult task to create such the shift schedule of short time workers in the large-scale HIC in Japan, because there are many problems for creating it[16].

The author has been doing research on such the shift scheduling by using Cooperative Evolution (CE)[9, 10, 11, 12, 13, 14, 16]. In the original GA[1], each individual evolves in a competitive fashion, whereas individuals in a populations evolve in cooperation with each other in CE, and the entire population improves. There are several research works dealing with a Nurse Scheduling Problem (NSP) [2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14] as similar scheduling problem. NSP can be reduced to the symbol combination optimization problem, because three shift system is adopted in most of the General Hospital in Japan. However, the issue handled in this paper can not be simplified by the same manner, because it contain several aspects. Then the technique for NSP can not be directly applied to the scheduling handled in this paper. And several problems has been found in carrying out this research. Therefore, this paper proposes techniques to automatically create and optimize the shift schedule of short time workers in the large-scale HIC by using CE, to solve a problem that many worker's attendances concentrate in a specific time zone, and to a problem where many breaks are concentrated in a specific break time zone, and an effective mutation operators. The CE for this problem basically optimizes the schedule by using a crossover operator.

2 Cording and Evaluation of Shift Schedule of Workers in Large-Scale HIC

In this problem, the work placements are as follows; service counter (SC), center registers (CR1—CR7), center sacker (CS), pet shop registers (PR1—PR2), pet shop sacker (PS), material shop registers (MR1—MR2), horticulture shop register (GR), horticulture shop sacker (GS) and electric tool register (DR). At SC, CRn ($n = 1, 2, \cdots, 7$), CS, PRn ($n = 1, 2$), PS and MRn ($n = 1, 2$), workers of up to two, one, seven, one, two and one can be placed respectively. At any other places, workers of up to one person can be placed.

An example case that the employment time of day is 09 : 00 to 20 : 00 is shown in this paper. The minimum unit of work placement time is 30 minutes, we call this time unit (TU). Therefore, separating the employment time zone in 30 minutes, each of them is defined as a TU. Totally, 22 TUs are included in one workday. The manager defines a placement queue at each TU. The placement queue contains work places that the manager wants to preferentially arrange for each TU.

2.1 Expression of the Whole Monthly Schedule and Definition of Individual and Population

In this research, an individual and an individual group are defined as shown in Fig. 1. As shown in Fig.1, the daily work placement sequence of each workday is defined as an individual. The work placement of the worker m at TU j of day i is given by a priority number $a_{im}(j)$. The manager defines the placement queue $\mathbf{c}_{ij}(k)$ for each TU. The element $\mathbf{c}_{ij}(k)$ at TU j of day i contains the k-th order of the work placement.

$$\mathbf{c}_{ij} = [c_{ij}(1), c_{ij}(2), \cdots, c_{ij}(K_{ij})], \tag{1}$$

where K_{ij} denotes the depth of the placement queue $\mathbf{c}_{ij}$. Therefore, the placement queues at workday i is summarized as follows,

$$\mathbf{c}_i = [\mathbf{c}_{i1}, \mathbf{c}_{i2}, \cdots, \mathbf{c}_{iT}], \tag{2}$$

where T denotes the number of TUs of one workday, $\mathbf{a}_{im}(l)$, $(l = 1, 2, \cdots T)$ denotes a natural number of $\{1, 2, \cdots, T\}$. In the example case shown in this paper, T is equal to 22.

The individual is defined by a vector $\mathbf{a}_{im}$ as following equation,

$$\mathbf{a}_{im} = [a_{im}(1), a_{im}(2), \cdots, a_{im}(T)], \tag{3}$$

The value of the element $a_{im}(j)$, or the individual m on day i, is to be an index of the placement queue vector $\mathbf{c}_{ij}$. For example, the work placement p_{imj} of the worker m at the TU j on the day i is given by the following equation using the placement queue $\mathbf{c}_{ij}$,

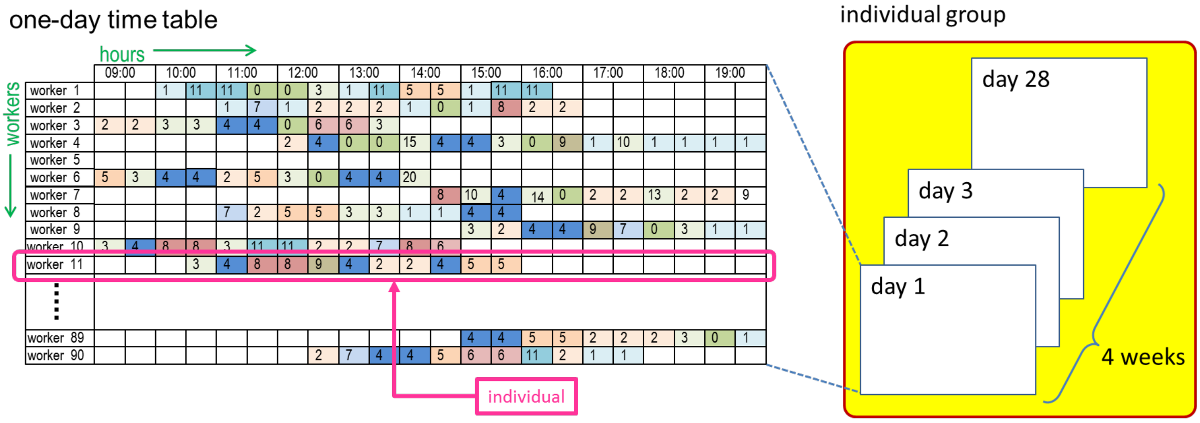
$$p_{imj} = \mathbf{c}_{ij}(a_{im}(j)) \tag{4}$$

The shift schedule on the day i is given by the following equation and coded as shown in Fig.1.

$$\mathbf{A}_i = [\mathbf{a}_{i1}, \mathbf{a}_{i2}, \cdots, \mathbf{a}_{iN}]^T \tag{5}$$

where N denotes the number of whole workers. The number in each box in Fig.1 shows the priority index of the placement queue $\mathbf{c}_{ij}(k)$ at the current TU. The whole shift schedule is defined by collecting all the one-day shift schedule $\mathbf{A}_i$ as follows,

$$\mathbf{S} = [\mathbf{A}_1, \mathbf{A}_2, \cdots, \mathbf{A}_N] \tag{6}$$



	09:00		10:00		11:00		12:00		13:00		14:00		15:00		16:00		17:00		18:00		19:00	
worker 1			1	11	11	0	0	3	1	11	5	5	1	11	11							
worker 2					1	7	1	2	2	2	1	0	1	8	2	2						
worker 3	2	2	3	3	4	4	0	6	6	3												
worker 4							2	4	0	0	15	4	4	3	0	9	1	10	1	1	1	1
worker 5																						
worker 6	5	3	4	4	2	5	3	0	4	4	20											
worker 7												8	10	4	14	0	2	2	13	2	2	9
worker 8					7	2	5	5	3	3	1	1	4	4								
worker 9													3	2	4	4	9	7	0	3	1	1
worker 10	3	4	8	8	3	11	11	2	2	7	8	6										
worker 11				3	4	8	8	9	4	2	2	4	5	5								
⋮																						
worker 89													4	4	5	5	2	2	2	3	0	1
worker 90								2	7	4	4	5	6	6	11	2	1	1				

Figure 1: Definition of an individual and an individual group. The number in the box denotes an index for the work placement queue defined by the manager. The work placement queue includes work placements is placed in the order in which the manager preferentially arrange for each TU.

2.2 Evaluation of the Individual and the Population

In this problem, there are several strong constraints. Working hours of each worker must not be interrupted. The work start time and work end time of each worker must be within the employment time. When the initial population is generated, the value of each individual is not assigned a value exceeding the size of the work placement queue, $\mathbf{c}_{ij}$. Therefore, no element of individual has value exceeding the size of the work placement queue. On the other hand, there are many week constraints in this problem. In this research, these week constraints are expressed in penalty functions. Each individual and the whole population is evaluated by 22 penalty functions as shown in [16] in detail. The smaller these penalty functions, the better the individual or the population.

These 22 penalty functions are simply scalarized as shown by the following equation,

$$E(g) = \sum_{k=1}^{22} F_k. \tag{7}$$

The population is evaluated by the function $E(g)$ at the generation g.

3 Optimization by CE with Several Special Operators

This paper proposes an algorithm of the CE specially for optimizing the shift schedule of the short time workers in the large-scale HIC in Japan. As shown in Fig.2, the CE optimizes the population basically by using a crossover operator. A mutation operator is activated when certain condition is met. In the optimization process, there are some TUs where the number of workers is remarkably small. In order to solve this problem, a worker number adjustment operator is applied in the CE. On the other hand, there would be specific TUs where many breaks are also concentrated in the optimization process. In order to solve this problem, a break time adjustment operator is applied in the CE.

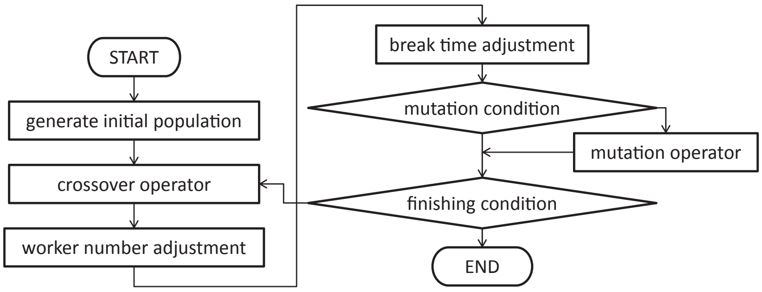


Figure 2: Optimization flow of the cooperative GA for the scheduling of the large-scale HIC.

3.1 Crossover Operator

The CE optimizes the individuals and the population basically by using the crossover operator as shown in Fig. 3. One workday giving large evaluation value is firstly selected by roulette selection. Then two workers in the workday are selected as parents. One of these parents as giving large evaluation value is selected by a roulette selection manner. Another one is randomly selected. And then, two-point crossover is applied to these parents. Three pairs of child individuals are generated by the two-point crossover. The child pairs are once returned to the shift schedule. Whole population is evaluated by the function E. The population giving the smallest value of E is selected at this time. This operation is repeated ten times. The population giving the best evaluation finally given is preserved to the next generation.

3.2 Worker Number Adjustment Operator

In the process of the optimization, there may be a population with several TUs that number of workers exceeds by means of the crossover operator. Such TUs are identified and the number of workers at the TU must be adjusted. A worker number adjustment operator identifies such TUs and adjusts the number of workers at the TU as shown in Fig.4. In the example in the figure, worker number adjustment operator is looking for a TU that is short of workers from the left. When the operator finds a TU that is deficient in the number of workers, it looks for the next TU at the end or begin of one of the workers' working hours and adds the appropriate value there. The figure shows an example when the operator adds the appropriate value at the end of the working hours.

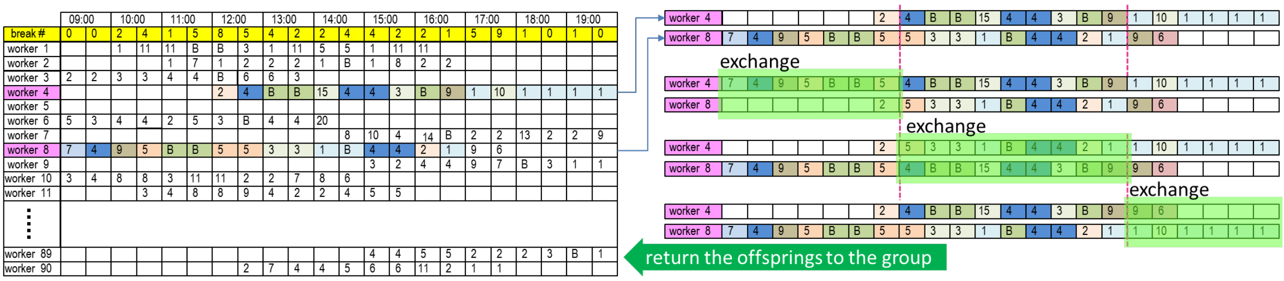
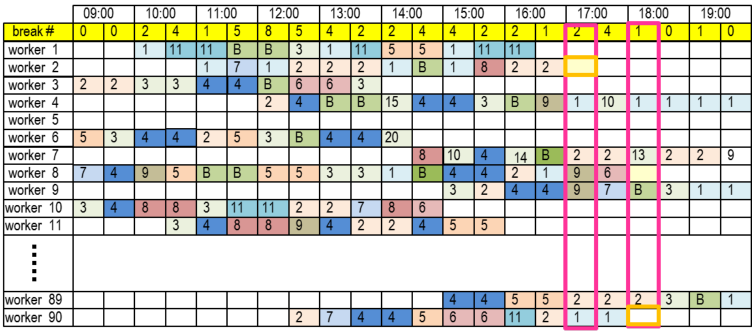


Figure 3: The crossover operator is applied as a basic optimizing operator. Two-point crossover is handled in this operator. By two-point crossover, three child pairs are generated.



	09:00		10:00		11:00		12:00		13:00		14:00		15:00		16:00		17:00		18:00		19:00	
break#	0	0	2	4	1	5	8	5	4	2	2	4	4	2	2	1	2	4	1	0	1	0
worker 1			1	11	11	B	B	3	1	11	5	5	1	11	11							
worker 2					1	7	1	2	2	2	1	B	1	8	2	2						
worker 3	2	2	3	3	4	4	B	6	6	3												
worker 4							2	4	B	B	15	4	4	3	B	9	1	10	1	1	1	1
worker 5																						
worker 6	5	3	4	4	2	5	3	B	4	4	20											
worker 7												8	10	4	14	B	2	2	13	2	2	9
worker 8	7	4	9	5	B	B	5	5	3	3	1	B	4	4	2	1	9	6				
worker 9													3	2	4	4	9	7	B	3	1	1
worker 10	3	4	8	8	3	11	11	2	2	7	8	6										
worker 11				3	4	8	8	9	4	2	2	4	5	5								
⋮																						
worker 89													4	4	5	5	2	2	2	3	B	1
worker 90								2	7	4	4	5	6	6	11	2	1	1				

Figure 4: Worker number adjustment operator. This operator finds the orange boxes and make them attendance state.

3.3 Break Time Adjustment Operator

In the process of the optimization, there would be a population that break time concentrates at several TUs. Under such the circumstances, the break time zone must be dispersed. A break time adjustment operator disperses such the break time zones as shown in Fig. 5. The break time adjustment operator reduces the number of breaks at the TU where break time concentrates and allow the number of workers to be replenished. The number of breaks at the selected TU is decreased and the number of workers is replenished.

Procedures of the break time adjustment operator is as follows; (1)The operator searches a TU that the number of workers less than the specified number is insufficient and also there are many breaks, the example is shown by the pink box. (2) One individual who takes break at the TU is randomly selected. (3) The selected break is moved to another TU in the individual determined probabilistically according to length among the work start time, break and the work end time. (4) The TU where the selected break is move to is determined as a TU including fewer number of the break time. (5)A random value, as a new priority number, is assigned to the TU where there has been a moved break time.

Even when such an operator is applied, there are cases in which it is not possible to suppress the concentration of the break time at the specific TU. Therefore, the following three kinds of improvements are proposed here. These improved operators are evaluated in numerical experiments to be described later.

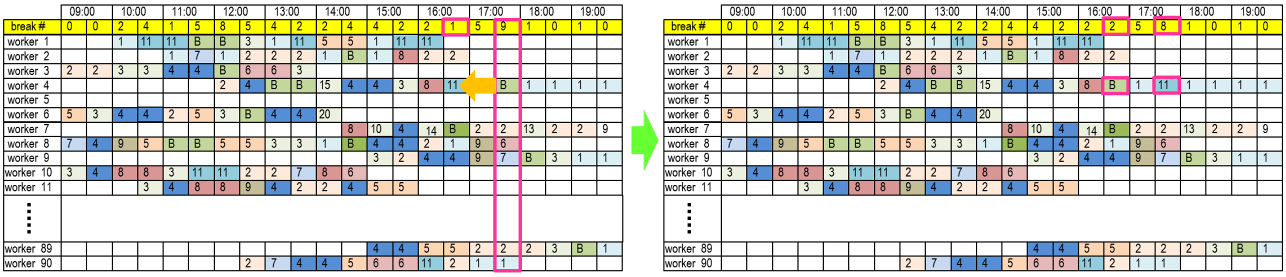


Figure 5: Break time adjustment operator.

- **•BTA1**: BTA1 operator randomly selects an individual including break at the selected TU that includes many break time. The break time is moved to another TU as shown above. BTA1 assigns a random number to the original TU.

•**BTA2**: BTA2 operator selects an individual giving the largest value of the penalty function f_{18}. This means that individuals giving small evaluation would not be changed. The selected break is probabilistically move to TU where gives larger value $e^i(m)$,

$$e^i(m) = \sum_{k=17}^{22} F_k^i(m), \tag{8}$$

where i denotes the workday, m denotes the worker and $F_k^i(m)$ denotes a value of the penalty function k of worker m at day i respectively. BTA2 assigns a random number to the original TU.

•**BTA3**: BTA3 operator also selects an individual giving the largest value of the penalty function f_{18}. The destination TU is defined in the same manner of BTA2. BTA3 operator assigns the lowest priority number to the original TU. The BTA3 operator is expected to have an effect of preventing the work placement of other workers from being changed. By means of this way, the work priorities at the TU is not required to be brought downward as much as possible.

3.4 Mutation Operator

The mutation operator provides a very effective search capability for the CE. In the case of CE, a mutation operator that mutates a small part of individuals and individual populations with low probability is required. However, when the mutation is applied for the optimization, there are a lot of issue to be considered. In this research, the following three types of the mutation operator are considered and tried in the later numerical examination.

MO1: MO1 randomly selects a workday and randomly selects a worker should be mutated and randomly changes the priority numbers of the workday. At this time, the work priorities at all the TU of the workday are required to be brought downward.

MO2: MO2 randomly changes the attendant time zone of a worker at a workday. The work priority of other worker must be brought downward or upward.

MO3: A workday when gives the larger value of evaluation is selected by means of roulette selection manner. MO3 randomly selects two workers. One-day time zone of these two workers are exchanged each other.

4 Numerical Experiments

Numerical experiments are executed based on conditions of the real large-scale HIC as follows. The shift schedule for four weeks is created. The number of registered short time workers is 50. The number of workers who can be assigned to SC is 12. The number of workers who can be assigned to the horticulture shop is 2. The number of workers who can be assigned to the material shop is 2. The number of new face workers is 6. The working hours is from 9:00 to 20:00. The number of TUs on a workday is 22. The finishing generation of the optimization is 3000. The mutation probability is 0.01. Ten times of the optimization are executed under each condition.

A computer environment applied to the numerical experiments are Windows 7 Home Premium SP1, Core i5 CPU (750 @2.67GHz, 2.79GHz), 4GB RAM, Visual Studio 2005 Standard Edition ver. 2.0.50727 SP2.

For comparing the break time adjustment operators, the following numerical experiments are executed without the mutation operator.

(E1) experiments without the break time adjustment operator

(E2) experiments with the break adjustment operator before improvement

(E3) experiments with BTA1

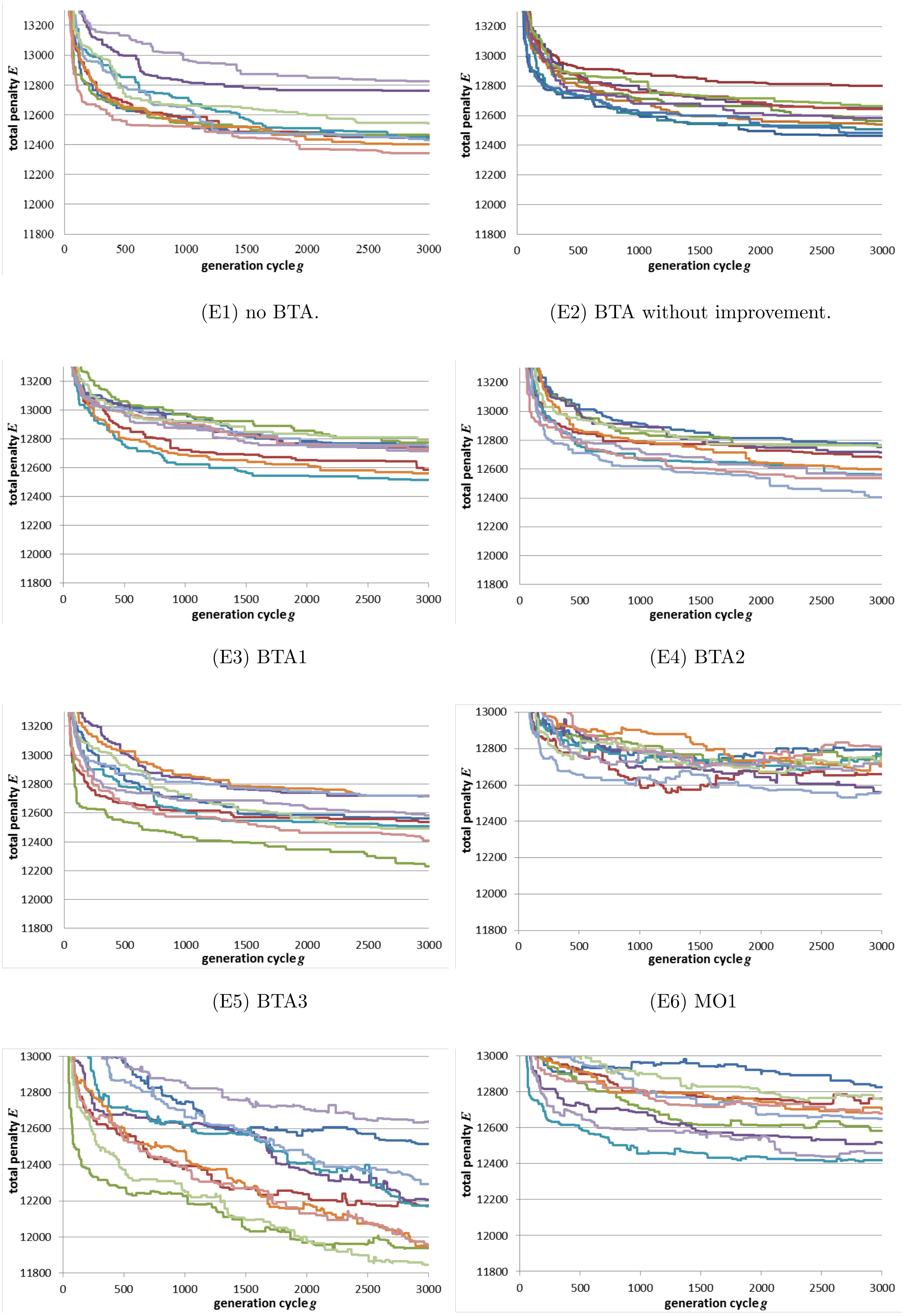
(E4) experiments with BTA2

(E5) experiments with BTA3

The progresses of the optimization are shown in Fig.6.

Table 1 shows result of comparison of the break time adjustment operators. In Table .1, ”without” denotes the cases without the break adjustment operator and ”BTA” denotes the break adjustment operator. The numbers in table show the number of breaks at each TU. As shown in Figs.6 (E1)-(E5) and Table 1, it can be said that the BTA3 shows the best result.

Finally, the following experiments are executed for comparing the mutation operators.



(E1) no BTA.

(E2) BTA without improvement.

(E3) BTA1

(E4) BTA2

(E5) BTA3

(E6) MO1

(E7) MO2

(E8) MO3

Figure 6: Numerical experiment results. A comparison among the break adjustment operators and a comparison among the mutation operators. The horizontal axis and the vertical axis of each graph represents the number of generation cycles of the optimization g and total evaluation value E respectively.

(E6) experiments with MO1

(E7) experiments with MO2

(E8) experiments with MO3

The BTA3 is applied to these experiments. As shown in Figs.6 (E6)-(E8), the MO2 shows the best result of the optimization.

Table 1: A comparison of the number of TUs for the number of breaks per each TU.

break # per TU	without	BTA1	BTA2	BTA3
1 break	63	57	56	57
2 breaks	105	113	103	86
3 breaks	102	116	127	127
4 breaks	87	110	110	116
5 breaks	68	74	75	66
6 breaks	29	13	15	20
7 breaks	10	6	0	4
8 breaks	3	0	0	0
9 breaks	1	0	1	0

5 Conclusion

This paper has proposed the cooperative evolutionary technique, it is called as CE, to automatically create the shift schedule of the short time workers in large-scale HIC in Japan. This CE method cooperates among individuals to search the entire population unlike ordinary GA. The population expresses the shift schedule. Each individual expresses a one-day schedule of one worker. In this research, the specialized crossover operator, the work number adjustment operator, the break time adjustment operator and the specialized mutation operator are proposed for CE. The numerical experiments shows effectiveness of these operators.

Acknowledgment

This work was supported by JSPS KAKENHI Grant Number JP17K00339.

The author, Dr.Ohki, would like to thank to her cats Blackin', Blanc, Caramel, Civita, Marron, Yukichi, Malkin and Ojarumaru for giving her the healing daily life.

The author, Dr.Ohki, would like to thank to her lesbian partner, Miss Tomo Toku, for supporting her daily living and research work.

References

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

[2] I.Berrada, J.A.Ferland, 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).

[3] E.K.Burke and P.Cowling, A Memetic Approach to the Nurse Rostering Problem, Applied Intelligence 15, pp.199-214 (2001).

[4] B.Cheang, H.Li, A.Lim and B.Rodrigues, Nurse Rostering Problems - a bibliographic survey, European Journal of Operational Research 151, pp.447-460 (2003).

[5] J.F.Bard and H.W.Purnomo, Preference Scheduling for Nurses using Column Generation, Europiean Journal of Operational Research 164, pp.510-534 (2005).

[6] 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)

[7] F.D.Croce and F.Salassa, A Variable Neighborhood Search based Matheuristic for Nurse Rostering Problems, Proc of 8th International Conference on the Practice and Theory of Automated Timetabling, pp.167-175 (2010).

[8] F.D.Croce and F.Salassa, A Variable Neighborhood Search based Matheuristic for Nurse Rostering Problems, Annals of Operations Research, pp.185-199 (2014).

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

[10] Makoto Ohki, Effective Mutation Operator for Nurse Scheduling by Cooperative GA and Its Parallel Processing, Proc. of 19th Int. ACM Workshop on Parallel Architectures and Bioinspired Algorithms, pp.1-8 (2010).

[11] Makoto Ohki and Hideaki Kinjo, Penalty Weight Adjustment in Cooperative GA for Nurse Scheduling, Proc. of IEEE 2011 Third World Congress on Nature and Biologically Inspired Computing, pp.76-81, (2011).

[12] Makoto Ohki, Effective operators using parallel processing for nurse scheduling by cooperative genetic algorithm, Proc. of International Journal of Data Mining, Modelling and Management, pp.57-73 (2012).

[13] Makoto Ohki, Periodic Mutation Operatorfor Nurse Scheduling by Using Cooperative GA, Proc. of International Journal of Applied Evolutionary Computation, Issue 3, pp.1-16 (2012).

[14] Makoto Ohki and Satoru Kishida, An Effective Nurse Scheduling by a Parameter Free Cooperative GA, Proc. of 17th European Conf. EvoApplications 2014, pp.955-966 (2014).

[15] Makoto Ohki, A Parameter Free Nurse Scheduling, Proc. of 2015 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE), DOI: 10.1109/FUZZ-IEEE.2015.7337897 (2015).

[16] Makoto Ohki, An effective cooperative GA for shift scheduling of short time employees in large-scale home improvement retailer, Proc. of 22nd International Conf. on Soft Computing Evolutionary Computation, Genetic Programming, Swarm Intelligence, Fuzzy Logic, Neural Networks, Chaos, Bayesian Methods, Intelligent Image Processing, Bio-Inspired Robotics (MENDEL 2016), pp.1-8 (2016)