Satisfactory Design of Cogeneration System using Genetic Algorithm

Satoshi Hirai
Graduate School of Engineering,
Doshisha University
1-3 Tatara Miyakodani
Kyotanabe, Kyoto 610-0394
Email: hirai@mikilab.doshisha.ac.jp

Tomoyuki Hiroyasu
Department of Engineering,
Doshisha University
1-3 Tatara Miyakodani
Kyotanabe, Kyoto 610-0394
Email: tomo@is.doshisha.ac.jp

Mitsunori Miki
Department of Engineering,
Doshisha University
1-3 Tatara Miyakodani
Kyotanabe, Kyoto 610-0394
Email: mmiki@mail.doshisha.ac.jp

Hisashi Shimosaka
Graduate School of Engineering,
Doshisha University
1-3 Tatara Miyakodani
Kyotanabe, Kyoto 610-0394
Email: hisashi@mikilab.doshisha.ac.jp

Yoichi Tanaka
Toho Gas Co.,Ltd.
507-2 Sinpoumachi
Tokai, Aichi 476-0005
Email: y_tanaka@tohogas.co.jp

Syuichi Aoki
Toho Gas Co.,Ltd.
507-2 Sinpoumachi
Tokai, Aichi 476-0005
Email: saoki@tohogas.co.jp

Yoshito Umeda
Toho Gas Co.,Ltd.
507-2 Sinpoumachi
Tokai, Aichi 476-0005
Email: umeda@tohogas.co.jp

***Abstract*— This paper introduces the optimum design of Cogeneration System (CGS) using the Genetic Algorithm (GA). CGS is the energy reusing system which generates more than two energies from one energy source. To design CGS, the types of machines and load scheduling should be determined. However, the optimum design of CGS is too complicated even for the Expert. One of the solutions for this problem is using GA. GA is the optimization model imitating evolution of life. If the coding of the problems is proper, GA can be applicable to many problems. However, proper coding for the problems is difficult, especially for CGS, because it has three different design variables which consist of integer values and real values. To discuss the effective coding, this paper considers four models. First is simplest coding model. Second is two-step optimization model with integer coding. Third is two-step optimization model with the integer coding and the penalty method. Last is three-step optimization model with the integer. As a result of the experiments, three-step optimization model could achieve the higher energy efficiency design of CGS than the expert.**

I. Introduction

Recently, Cogeneration System (CGS)[1] has been paid attention, because CGS generates the energies effectively. In general, CGS means a continuous production of more than two energies from a single energy source. The key issue in the CGS design is the selection of proper machines and load factor per each time in plural energy demands. If CGS works under the proper machines and load factor, it can supply target energies from less energy source. However, CGS design is generally very complex and a very difficult problem. Conventionally, CGS design was based on the experience and technical instinct of design projector, and it was researched by using Hamiltonian Algorithm (HA)[2].

On the other hand, Genetic Algorithm (GA)[3] is an optimization technique based on natural genetics. GA is versatile optimum model which is applicable to many actual problems. For example, there are the optimum control of a Flexible Manipulator [4], and the Prediction of Protein Tertiary Structures [5] and so on. GA shows great results among them. Therefore, it is expected that GA is effective for CGS design problem too. This research examines the optimization with adopting GA to CGS design problem.

The coding for the problem is the key issue of optimization by GA. Especially, CGS design problem has various design valuables, such as Operation Time, the Number and Type of Machine, and the Load Factor per each time. CGS design problem consists of two factors. One is compound optimization problem searching proper selection of machines. Another is the continuous problem searching proper load factor in each machine. Therefore, designing effective coding for the CGS design problem is very difficult. This paper proposes effective coding for CGS design problems, and optimum design of CGS supplying energies from less energy source.

II. Cogeneration System

A. Overview of CGS

CGS consists of Generator, Exhaust Heat Unit and Boiler. These machines generate Electricity, Air Conditioner, Heater and Hot water. The characteristic of each machine is explained as follows:

- **Generator**
  Generator generates Electricity and discharge Heats at the same time. The kinds of Heats are Gas, Steam and Boiling Water. Machine Code decides Maximum output and Generating efficiency of Electricity and Heat. Conventionally, these Heats generated by Generator was exhausted. CGS can achieve high energy efficiency rate by reusing Heat.
- **Exhaust Heat Unit**
  Exhaust Heat Unit reuses Heat (Gas, Steam, Boiling Water) generated from Generator, and generates Energies such as Heater, Air Conditioner and Hot Water.
- **Boiler**
  Boiler generates Heats (Steam and Boiling Water) when Heats from Generator isn't enough to generate Energies (Heater, Air Conditioner and Hot Water).

Heats and Energies depend on Machine Code which decides the characteristic of Machine. One Machine cannot generate more than two Heats or Energies. And Machine Code decides Maximum output and Generating efficiency of Electricity and Heat. Conventionally, these Heat generated by Generator was exhausted. CGS can achieve high energy efficiency rate by reusing Heat.

If the selection of these machines and Load Factor in each machine is proper, CGS can generate the target energy demand from less energy resource than conventional power plants. Overview of the CGS is shown Fig.1.

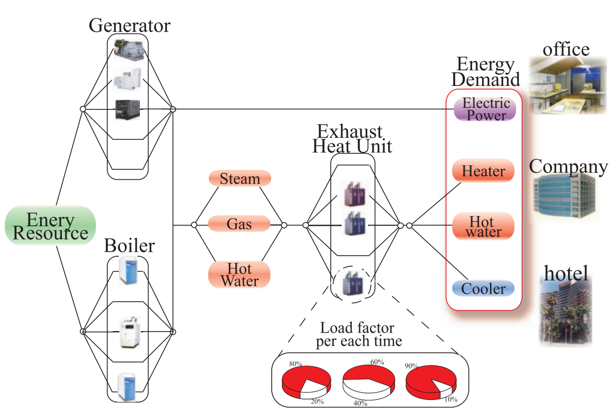

Fig. 1. Overview of the CGS

B. The design variables of CGS design

In CGS design problem, proper selection of Machine and Load Factor per each time should be determined. Thus, there are three kinds of design variables which are Operation time, Machine Code and Load Factor. Operation Time and Machine Code are integer value, and Load Factor is real values. The explanation of design variables is as follows,

- **Operation Time**
  In actual operation of CGS, it cannot start and stop many times during short hour. CGS design problem has to work each machine from operation start time to end time as the constraint.
- **Number and Type of Machine**
  The characteristic of Generator, Exhaust Heat Unit and Boiler depends on several Machine Codes which decide Maximum output, energy efficiency, the kind of generating Heats and Energy. Therefore CGS design problem needs to decide Number and Type of Machine.
- **Load Factor**
  Load Factor shows the operating rate of machines per each time.

III. Genetic Algorithm

Genetic Algorithm (GA) have been developed in an attempt to imitate the mechanics of the selection process in natural genetics [3]. GA combines selection, crossover, and mutation operators with the goal of finding the best solution to a problem. GA searches for this optimal solution until a specified termination criterion is met.

This research adopts Distributed Genetic Algorithm (DGA)[6] in CGS design problem. DGA consists of several subpopulations evolving separately and concurrently with occasional migration of individuals between subpopulations. DGA has high searching ability.

IV. How to apply GA to CGS design problem

This research examines optimization of CGS design problem using GA. The key issues for CGS are following:

- **Expression of gene.** CGS consists of compound problems whose design variables are integer and real values.
- **Way of crossover.** CGS has strong dependences between design variables and simple coding makes chromosome length long.

This section shows the expression of gene, treatment of individuals out of restriction, and gradual-step optimization model.

A. Expression of gene

In CGS design problem, searching ability depends on the expression of gene, because of compound problems of both integer values and real values. Our research proposes the simplest optimization model and Integer-Coded GA.

1) The simplest optimization model: The simplest optimization model is the simplest coding model. It expresses both of Machine Code and Load Factor design variable into the same chromosome. Machine Code is expressed by four bits. Load Factor is expressed by ten bits per each hour. For example, if there are fifteen machines, and objective time is one hour, the chromosome length becomes 210 bits. If objective time is ten hour, it becomes 1,560 bits. So, this model has the feature that chromosome length becomes long by depending on time. This model does not optimize Operation Time. The start time is set up with first time of objective demands, and the end time is set up with last time of objective demands. Fig.2 shows the overview of the simplest optimization model.

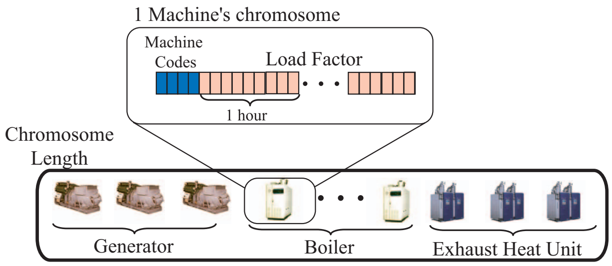

Fig. 2. Overview of the Simplest Optimization Model

2) Integer-Coded GA: It is thought that the simplest optimization model cannot achieve better searching ability. The reason is that the genetic operator of this model cannot work well, because the different design variables are expressed by the same chromosome. Therefore, our research proposes Integer-Coded GA (ICGA). ICGA is expressed both Operation Time and Machine Code by integer, because these design variables express integer. It is expected that ICGA can achieve better searching ability, because of crossover between design variables. Fig.3 shows overview of ICGA.

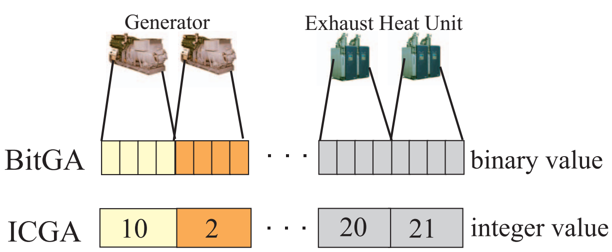

Fig. 3. Integer-Coded GA

B. The operation of constraints of CGS

1) Penalty method: In our research, Machine Code has the pattern of machine termination. When the selection of machines does not choose any code, energy demands are purchased from other electric power companies. This evaluation value is inferior to that of the best design in the real world. However, it shows the better value in the simulation. This leads to converge into local optimal solution in early stage of GA. Therefore, penalty is added to individuals including no machine, in order not to survive next generation. Penalty method is that penalty is added to individuals including lethal gene, and decreases this fitness. When Machine Code selects the medium electric power, total amount of energy source calculates equation 1. f in equation 1 indicates the total amount of energy source when CGS is not used.

$$\phi = f + 1000[kw] \tag{1}$$

C. Gradual-step optimization model

If all design variables are optimized at the same time, it takes long time, because there are three difference design variables. Therefore this research proposes optimization models which gradually optimize in each design variable.

1) Two-step optimization model: Firstly, this model optimizes selection of machine under the constant load factor. Then, load factor is optimized using optimum machines. It is expected that this model can achieve better searching ability in spite of dependence between design variables. Fig.4 shows the overview of two-step optimization model.

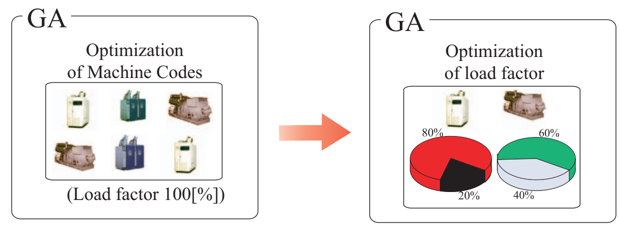

Fig. 4. Overview of the two-step optimization model

D. Three-step optimization model

Three-step optimization model is optimization model which optimizes Operation Time, Machine Code, and Load Factor one by one by using different GA. Firstly, Operation Time is optimized. At that time, small-scale GA determines Machine Code using 100% constant Load Factor. After the optimization, using optimized Operation Time, the different GA strictly determines Machine Code. Lastly, using optimized Operation Time and Machine Code, different GA strictly optimizes Load Factor. Fig.5 shows the overview of three-step optimization model.

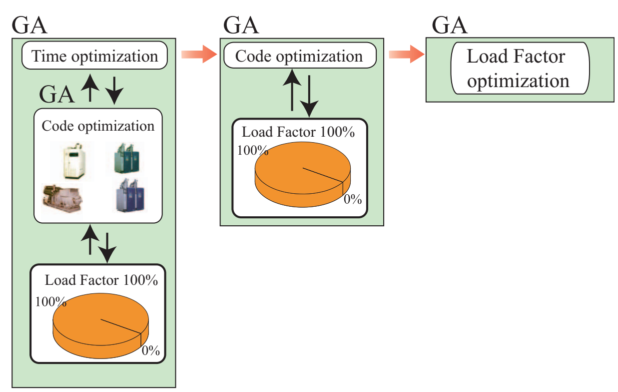

Fig. 5. Overview of the three-step optimization model

V. Experiment

A. Target energy demand

In this research, the target energy demands are shown in Fig.6, Fig.7, and Fig.8. The energy demand A of Fig.6 consists of constant Electric Power and Air Conditioner. For this energy demand, the best design of CGS is explicit. Therefore it can be used in order to check whether optimization model can achieve best solution or not. Fig.7 and Fig.8 are the demand of small office in summer. They consist of Electric Power, Air Conditioner and a little amount of Hot Water. The energy demand B does not change so much. On the other hand, C changes rapidly. Expert design is not the best, but better design. In the expert design, one Generator and one Exhaust Heat Unit are used.

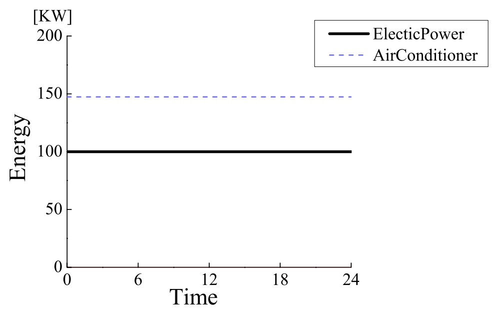

Fig. 6. Energy demand A

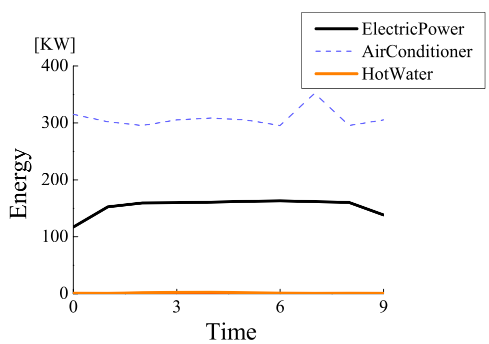

Fig. 7. Energy demand B

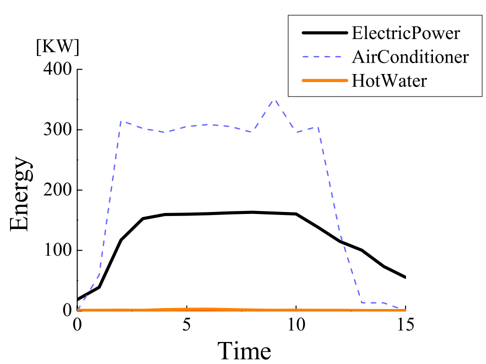

Fig. 8. Energy demand C

B. Experiment by simplest optimization model

The simplest optimization model is verified using energy demand A whose optimum design of CGS is explicit. Parameter of GA is shown in Table.I.

TABLE I

DEFAULT PARAMETER OF GA

Number of islands	1, 2, 5
Population size	10, 20, 50
Chromosome length	165
Number of evaluations	100,000
Crossover rate	1.0
Corssover method	2 point crossover
Mutation rate	0.0061
Selection method	Tournament
Tournament size	4
Elite individuals	5
Migration interval	5
Migration rate	0.5

Fig.9 is the calculation results by the simplest optimization model using the energy demand A. Fig.9 is the history of median values in thirty trials. "Without CGS" indicates evaluation value of purchasing from other electric companies. "Best" means the best evaluation value. The result is better when value of vertical becomes small.

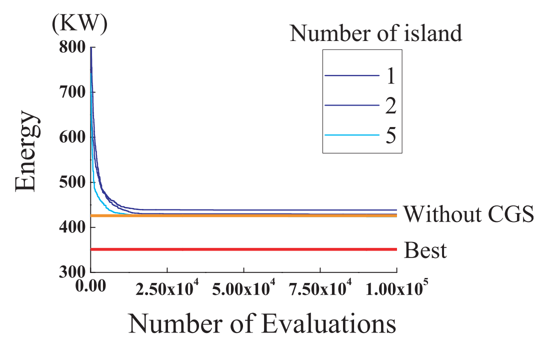

Fig. 9. History of median of evaluation values

As a result of experiment, the simplest optimization model has not achieved the best solution. The reason of this problem is that child individuals cannot inherit better characters of parent individuals in the genetic operator such as crossover and mutation. It is happened because there are design valuables of both Machine Code and Load Factor in the same chromosome.

1) Experiment by ICGA and two-step optimization model: The ICGA is verified using energy demand A whose optimum design of CGS is explicit. This experiment uses two-step optimization model which expresses Machine Code and Load Factor in different individuals. Table.II shows the parameters of GA, and the others are the same as these of Table.I. Load Factor fixes 100% in optimization of Machine Code.

TABLE II

PARAMETERS OF THE TWO-STEP GA

	Code	Load factor
Chromosome length(ICGA)	15	1350
Chromosome length(BitGA)	60	1350
Number of generation	400	10,000
Number of islands	2	1
Population size	20	10

Fig.10 is the calculation results of fifty trials by ICGA and two-step optimization model using the energy demand A. ”BitGA” in the figure indicates the result of bit coding, and ”ICGA” shows the result of ICGA. Both ”Without CGS” and ”Best” are the same as these of Fig.10.

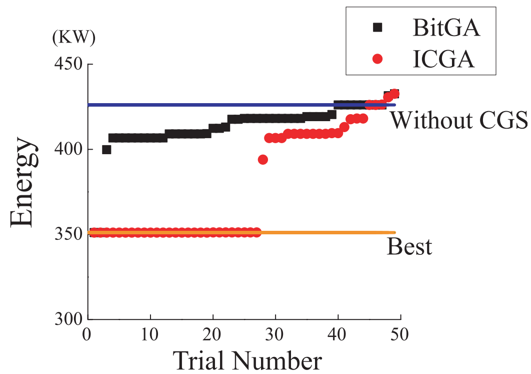

Fig. 10. The results of energy consumption using energy demand A

As the results of experiment, both of BitGA and ICGA have searched optimum design of CGS. Therefore, it is effective to adopt two-step optimization model which separates Machine Code and Load Factor. Moreover, compared with ICGA and BitGA, many trials of ICGA have searched optimum design of CGS. Therefore, the reliability of solution of ICGA is higher than that of BitGA. Therefore, ICGA can solve CGS design problem effectively.

Fig.11 is the calculation results in fifty trials using the energy demand B. The vertical line and horizontal line are the same as those of Fig.11. ”Expert” in the figure indicates total amount of consumption energy required by expert design.

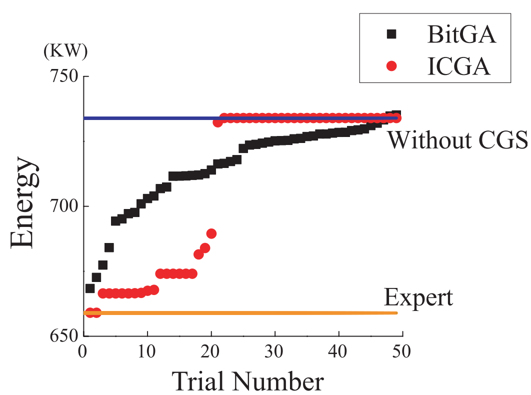

Fig. 11. The results of energy consumption using energy demand B

As the result of experiment, many trials of ICGA have searched optimum design of CGS than BitGA. However, many trials have fallen into ”Without CGS” which is the local optimum solution using medium electric power. In this case, the difference of total amount of energy source between ”Without CGS” and improper Machine Code is large. Then, the convergence is occurred in the early stage of search of GA.

C. Experiment by penalty method

The ICGA and two-step optimization model with penalty method is verified using energy demand C. Parameters of GA in the experiment is the same as those of Table.II.

Fig.12 is the result of fifty trials. Result of the penalty method shows ”ICGA with penalty method” in Fig.12

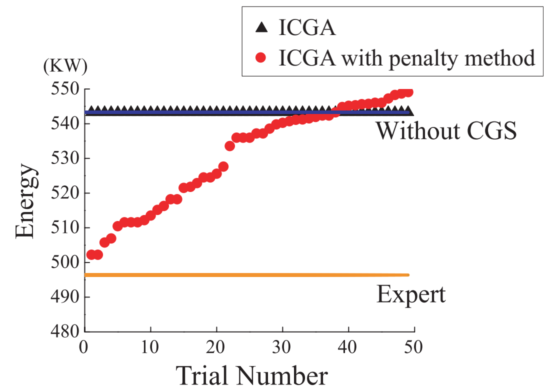

Fig. 12. The results of energy consumption using energy demand B

Fig.13 indicates Machine Code and Load Factor after the optimization. The vertical line in the figure shows the load factor in each time, horizontal line shows hour. The meshed area of the graph shows the minimum load factor of each machine. When load factor is lower than this area, machine is automatically stopped.

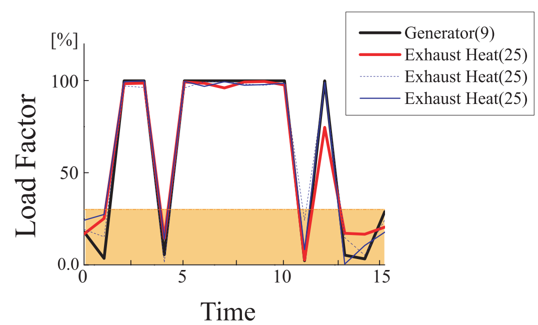

Fig. 13. The result of Machine Code and Load Factor

As the result of experiment, optimization model not using penalty method has fallen into local minimum solution. On the other hand, although the results of penalty method are inferior to the design by expert, it has achieved the better results. However, according to Fig.13, the design of CGS by GA is improper in actual use of CGS. The design of GA has started and stopped CGS three times during fifteen hours. The actual CGS cannot be stopped many times in a short hour, because huge load is needed to start CGS.

D. Experiment by three-step optimization model

The three-step optimization model is verified using energy demand C. Parameters of GA show in Table.III. The other parameters are the same as Table.I. The CGS design problem has many local minimum solutions. The number of island is set to five from two. This setting can keep the diversity of GA and avoid falling into the local minimum solution.

TABLE III

PARAMETERS OF THE THREE-STEP GA

	Time	Code	Load factor
Number of islands			5
Population size			50
Chromosome length	2	15	2400
Max of generation	20	200	5000

Fig.14 is the calculation results in thirty trials using the energy demand C. Fig.15 and Fig.16 shows the relation among energy demand and Machine Code and Load Factor of Generator and Exhaust Heat Unit by GA and Expert.

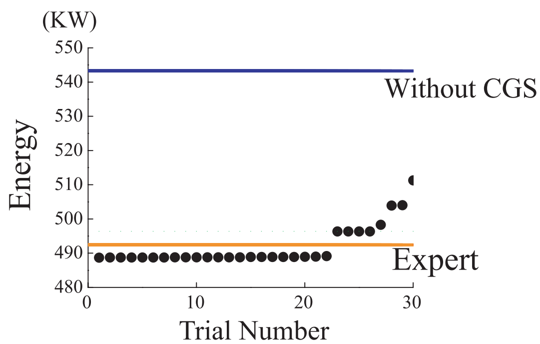

Fig. 14. The results of energy consumption using energy demand C

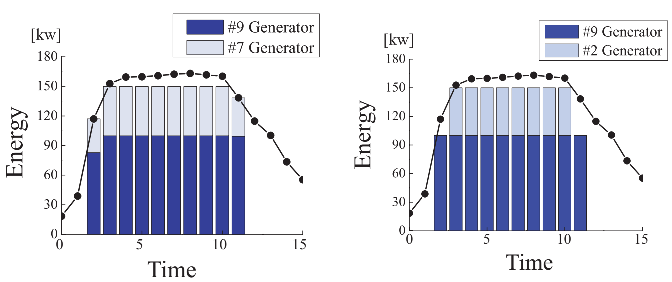

Fig. 15. The result of Machine Codes and Load Factors of Generator(left:GA, right:Expert)

As the result of experiment, many results of three-step optimization model have designed better CGS than that of expert. Therefore, it is effective to use three-step optimization model that separates Operation Time and Machine Code and Load Factor. Moreover, according to the comparison between CGS designs by Expert and GA, the design by GA has been different from the expert. It is thought that GA design has been more effective. As for Load Factor, GA can adjust proper Load Factor per time. On the other hand, expert cannot. Therefore, GA has designed CGS efficiently.

VI. CONCLUSION

This paper has examined the proper coding for CGS design. The key issue in CGS design problem is how to apply

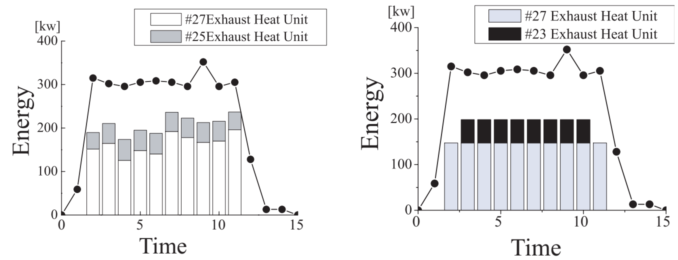

Fig. 16. The result of Machine Codes and Load Factors of Exhaust Heat Unit(left:GA, right:Expert)

GA to CGS, because there are three design variables that are Operation Time, Machine Code, and Load Factor. This research considers following two points:

- **Expression of genes.**
- **Way of crossover.**

We have proposed two coding methods that are the simplest optimization model and ICGA. The simplest optimization model expresses design variables by the same genes. ICGA uses integer coding for the design variables which are integer. This research also proposed gradual-step optimization model in order to consider the dependence between the design variables and express each design variable by different genes.

As a result of experiments, the simplest optimization model has not designed even for the simplest energy demand. However, when we used ICGA and gradual-step optimization model, even if target energy demand is very complex, these methods have designed better CGS than that of expert. Therefore, it has proved that GA was effective to solve CGS design problem.

REFERENCES

[1] R.Yokoyama; Itho.K; Akagi.S Determining the Optimal Equipment Capacities of CogenerationSystems *Trans. Jap. SOC. Mech. Eng.* 56-522,c(1990-2)

[2] M.Yanagi; T.Uekusa; J.Yamada; K.Shinjo; optimal design of cogeneration system by using Hamilatonian Algorihm *Proceedings of Building Simulation '99* Volume 2 : 699-706.

[3] D.E.Goldberg Genetic Algorithms in Search, Optimization and Machine Learning *Addison-Wasley Publishing Company* 1989

[4] N.M.Kwok, S Kwong:Control of a Flexible Manipulator Using a Sliding Mode Controller with Genetic Algorithm Tuned Manipulator Dimension *GECCO 2003* pp.2191-2202 (2003)

[5] Steven R.Michaud, Jesse B.Zydallis, Gary B.Lamont, Paul K.Harmer, Ruth Pachter:Protein Structure Prediction with EA Immunological Computation *GECCO 2001* pp.1367-1374 (2001)

[6] R,Tanese Distributed genetic algorithms. *3rd International Conference on Genetic Algorithms* 1989,434-439.