Double-deck Elevator System using Genetic Network Programming with Genetic Operators based on Pheromone Information

Lu Yu
Graduate School of Information, Production and Systems, Waseda University
Hibikino 2-7, Wakamatsu-ku, Kitakyushu, Fukuoka, 808-0135, Japan
yulu1006@fuji.waseda.jp

Jin Zhou
Graduate School of Information, Production and Systems, Waseda University
Hibikino 2-7, Wakamatsu-ku, Kitakyushu, Fukuoka, 808-0135, Japan
zhoujin@asagi.waseda.jp

Fengming Ye
Graduate School of Information, Production and Systems, Waseda University
Hibikino 2-7, Wakamatsu-ku, Kitakyushu, Fukuoka, 808-0135, Japan
yefengming@fuji.waseda.jp

Shingo Mabu
Graduate School of Information, Production and Systems, Waseda University
Hibikino 2-7, Wakamatsu-ku, Kitakyushu, Fukuoka, 808-0135, Japan
mabu@aoni.waseda.jp

Kaoru Shimada
Information, Production and Systems Research Center, Waseda University
Hibikino 2-7, Wakamatsu-ku, Kitakyushu, Fukuoka, 808-0135, Japan
k.shimada@aoni.waseda.jp

Kotaro Hirasawa
Graduate School of Information, Production and Systems, Waseda University
Hibikino 2-7, Wakamatsu-ku, Kitakyushu, Fukuoka, 808-0135, Japan
hirasawa@waseda.jp

ABSTRACT

Genetic Network Programming (GNP), one of the extended evolutionary algorithms was proposed, whose gene is constructed by the directed graph. GNP is distinguished from other evolutionary techniques in terms of its compact structure and implicit memory function. GNP can perform a global searching, but it lacks of the exploitation ability. Since the behavior of GNP is characterized by the balance between exploitation and exploration in the search space, we proposed a hybrid algorithm in this paper that combines GNP with Ant Colony Optimization (ACO). The genetic operators are operated using the pheromone information in some special generations. We applied the proposed hybrid algorithm to a complicated real world problem, that is , Elevator Group Supervisory Control System (EGSCS). The simulation results showed the effectiveness of the proposed algorithm.

Categories and Subject Descriptors

I.2.8 [**Artificial Intelligence**]: Problem Solving, Control Methods, and Search

General Terms

Algorithms, Performance

Keywords

Elevator Group Supervisory Control System, Genetic Network Programming, ant colony optimization, genetic operator, hybrid algorithm

1. INTRODUCTION

In the past, there has been increasing interest in imitating living things to develop powerful algorithms for difficult optimization problems. Genetic Network Programming (GNP)[1], a graph-based evolutionary method, has been proposed several years ago. GNP extends the biologically motivated Genetic Algorithm (GA), and relies on two genetic operators (crossover and mutation).

Although GNP has been applied to many applications, one of the main obstacles in applying GNP to complex problems is high computational cost due to its slow convergence rate. A common strategy for overcoming the GNP's slow convergence problem is to combine GNP with other techniques to balance the exploitation and exploration.

Our motivation in this paper is to introduce a positive feedback mechanism into GNP, therefore, we combined GNP with Ant Colony Optimization (ACO)[2,3], which enables the rapid search of a global solution. Generally speaking, traditional algorithms use the crossover and mutation operator to generate the next population without any information in the previous generations. Different crossover or mutation operators can make different evolutionary processes. Then, determining the type of crossover and mutation operators is very important for obtaining the good results. In this paper, the proposed algorithm is very different from combining GA and ACO[4]. The genetic operators have been done using the pheromone information of the nodes and branches, which are calculated by ACO. The new operations can be interpreted as a kind of exploitation in the structure of GNP.

Since Elevator Group Supervisory Control System (EGSCS) is similar to many other stochastic traffic control problems, it should find the strategy to minimize the service time and to maximize the elevator group transportation capacity. In previous studies, elevator system using GNP shows better

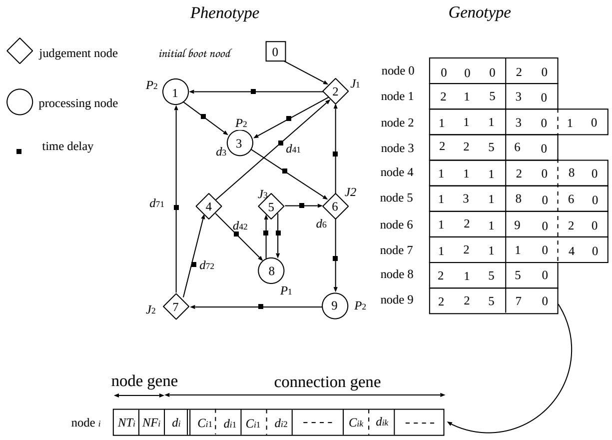


Figure 1: Basic Structure of GNP.

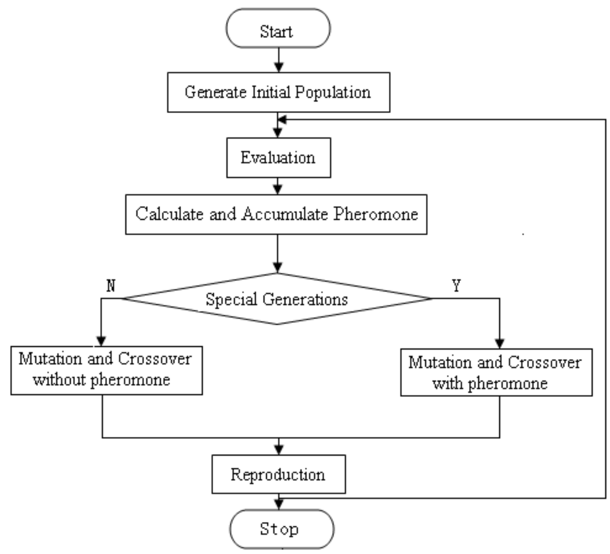


Figure 2: The flowchart of GNP with ACO.

performances than the conventional methods[5]. In this paper, GNP with ACO is proposed expecting to be appropriate for the assignment problem in elevator systems. The reason is that: GNP with ACO can realize rules based on EGSCS due to its directed graph structure, which makes EGSCS more flexible in different traffics. And also, the controller of EGSCS can be built by an evolutionary method with mutation, crossover and selection, which could develop new efficient and effective rules. Experimented results showed that the proposed algorithm has a faster convergence speed than those with conventional crossover and mutation operators, especially in the small population size.

The paper is organized as follows. Section 2 contains the description of the hybrid algorithm. Section 3 describes the application of the proposed algorithm to EGSCS. Section 4 shows the simulation conditions and results. Section 5 is devoted to conclusions.

2. ALGORITHM OF GNP WITH ACO

2.1 Genetic Network Programming (GNP)

Figure.1 shows the basic structure of GNP. GNP is composed of one start node and plural number of judgment nodes and processing nodes. The start node has no functions and no conditional branches. Each judgment node returns a judgment result and determines the next node to be executed. In processing nodes, actions are conducted to environments. The node transition begins from a start node, and there is no terminal node.

A GNP has an iterative procedure which maintains a population of candidate solutions. During each iteration step, the individuals in the current population are evaluated, and on the basis of these evaluations, a new population of the candidate solutions is formed. The initial population can be chosen randomly. In order to search other individuals in the search space, some variations are introduced into the new population by genetic operations. Then, the fitness of the new individuals are calculated until the terminal condition.

2.2 Ant Colony Optimization (ACO)

ACO is firstly proposed by Dorigo as a multi-agent approach to the difficult combinatorial optimization problems. Basically ACO uses two functions to guide the search toward the optimal solution when it is applied to the traveling salesman problem. Let $H^n(r,a)$ be the intensity of the pheromone on the section (r,a) of the trail at time n. After all ants have generated the tours, the pheromone intensity becomes updated as follows at time n

$$H^n(r,a) = (1-\rho)H^{n-1}(r,a) + \sum_{m \in M} h_m^n(r,a), \qquad (1)$$

where $\rho \in (0,1)$ is a parameter of evaporation. M is the set of suffixes of ants in the colony and $h_m^n(r,a)$ is the intensity of the pheromone of the section (r,a), which was laid by ant m at time n.

Let $\eta(r,a)$ be the visibility between the vertex r and a. $\eta(r,a)$ is the inverse of the distance of the section (r,a) in the traveling salesman problem. The probability that ant m chooses a as the next vertex is given by Eq.(2) when ant m is at the vertex r at time n

$$p_m^n(r,a) = \begin{cases} \frac{H^n(r,a)^\alpha \eta(r,a)^\beta}{\sum_{a \in V_m} H^n(r,a)^\alpha \eta(r,a)^\beta} & if \quad a \in V_m, \\ 0 & \text{otherwise.} \end{cases} \qquad (2)$$

where V_m is the set of vertices not visited yet by ant m, α and β are two parameters that control the relative importance of the trail versus visibility.

2.3 GNP with ACO

2.3.1 *Basic structure of GNP with ACO*

Combining GNP with ACO is proposed to improve the exploitation ability of GNP using the pheromone information, that is, using the transition and fitness information of GNP individuals. It converges on an optimal solution through the accumulated and vapored pheromone information.

Figure.2 shows the flow chart of the proposed GNP with ACO. The procedures of GNP with ACO can be summarized as follows:

- ***Initialization phase***: We first determine the number of nodes and connections in GNP.

- ***Pheromone calculation phase***: The pheromone information of each branch of GNP is calculated by using the fitness values and frequency of the transitions as follows

$$h^n_m(i,k,a) = \frac{F^n - f^n_m}{F^n} \cdot \gamma^n_m(i,k,a), \tag{3}$$

where,
$h^n_m(i,k,a)$: pheromone from the kth branch of node i to node a of individual m in the nth generation
f^n_m: fitness of individual m in the nth generation
F^n: the worst fitness in the nth generation
$\gamma^n_m(i,k,a)$: the frequency of the transitions from the kth branch of node i to node a of individual m in the nth generation

Also the pheromone information of each node of GNP is calculated by the pheromone of its branches.

$$h^n_m(i) = \sum_{k\in A(i)} \sum_{a\in A(i,k)} h^n_m(i,k,a), \tag{4}$$

$h^n_m(i)$: pheromone of node i of individual m in the nth generation
$A(i)$: set of suffixes of branches from node i
$A(i,k)$: set of suffixes of the nodes connecting from the kth branch of node i

- ***Pheromone updating phase***: The pheromone on each branch of GNP is updated by the accumulation and vaporization using the following pheromone update function

$$H^n(i,k,a) = (1-\rho)H^{n-1}(i,k,a) + \sum_{m\in M} h^n_m(i,k,a), \tag{5}$$

where,
$H^n(i,k,a)$: pheromone from the kth branch of node i to node a in the nth generation
ρ : parameter of evaporation
M : set of suffixes of individuals

- ***Genetic operation phase***: The general generation and special generation are defined during this phase. In general generation, genetic operators are carried out conventionally, where the pheromone is considered as one of the attributes of branches of GNP. In special generations, the new individuals are produced using the genetic operators with pheromone information.

<*Mutation*> The more pheromone on the branch is, the higher probability of taking the connection of the branch would appear in the new GNP individual. At a special generation, the offspring is produced by the following $P^n(i,k,a)$,

$$P^n(i,k,a) = \frac{H^n(i,k,a)}{\sum_{a\in A(i,k)} H^n(i,k,a)}, \tag{6}$$

where,
$P^n(i,k,a)$: probability of connecting the kth branch of node i to node a in the nth generation

<*Crossover*> By consider the pheromone on the nodes of each parent, the offspring is produced under the following rules. N_p offspring p and N_q offspring q are produced by parent g and h
If $h^n_g(i) \geq h^n_h(i)$
node i of offspring p is from node i of parent g
node i of offspring q is from node i of parent h
If $h^n_g(i) < h^n_h(i)$
node i of offspring p is from node i of parent h
node i of offspring q is from node i of parent g

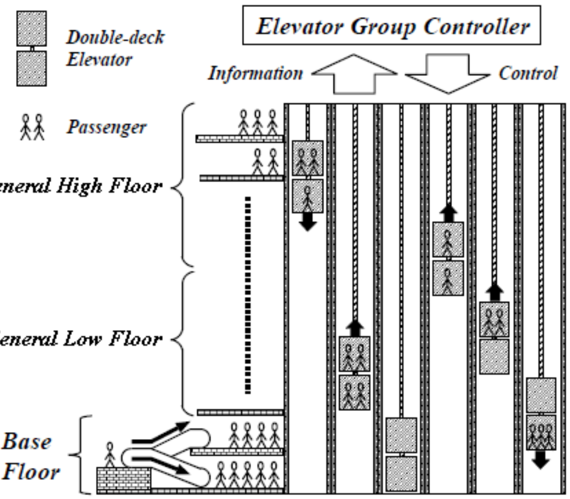


Figure 3: Outline of DDES.

- ***Termination phase***: The whole process is repeated until the terminal condition.

3. APPLICATION OF GNP WITH ACO

Efficient elevator group control is important for the operation of large buildings. In this paper, GNP with ACO is applied to the hall call assignment problem of Elevator Group Supervisory Control Systems (EGSCS) in order to show the effectiveness of the proposed method.

3.1 Review of EGSCS

Elevator Group Supervisory Control Systems (EGSCS)[6] are the control systems that systematically manage three or more elevators in order to efficiently transport passengers. Nowadays, Double-deck elevators with two cages being attached are proposed as the next generation system. Figure.3 shows the outline of DDES[7]. It allows the passengers on two consecutive floors to use the elevator simultaneously, significantly increasing the passenger transportation capacity of an elevator shaft. Meanwhile, Destination Floor Guidance System (DFGS) is installed in DDES, which is an elevator system where passengers can register their destination floors directly at elevator halls. In addition, One Cage Service is one of the specific features of DDES, where one cage stops without any service while the other cage serves passengers at the floor. Considering such features, DDES becomes more complex in their behaviors than conventional systems.

3.2 DDES with DFGS using GNP with ACO

It is difficult for a very large-scale stochastic system to select a suitable elevator for the following reasons. First, the controller must consider the hall calls which will be generated in the near future. Second, it must consider many uncertain factors, such as the number of passengers at the

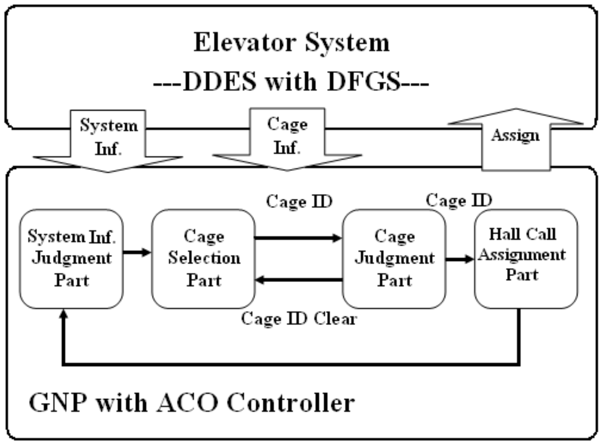


Figure 4: Structure of DDES with DFGS using GNP with ACO.

floors where the hall calls and destination registration are generated. Third, it must balance the items such as waiting time, long waiting time and so on.

The structure of DDES with DFGS using GNP with ACO is shown in Figure.4. It includes Elevator System and GNP with ACO controller which consists of four parts. The information is transferred through those parts.

3.2.1 Evaluation Items

In our proposed method the following 12 evaluation items are defined to construct GNP considering the features of DDES with DFGS.

AT_{sd} : Predicted arrival time of the assigned hall call to the self cage including the incremental arriving time of the already registered hall calls to the self cage
AET_{sd}: Maximum of the predicted arrival time plus elapsed time of the already registered hall calls since their assignment to the self cage
NP_{sd} : Number of passengers in the self cage
NC_{sd} : Number of assigned hall calls to the self cage
RR_{sd} : Predicted riding rate (number of passengers/cage capacity) of the self cage when the self cage arrives at the assigned hall call including the incremented riding rate of already registered hall calls to the self cage
CHC_{sd}: Check whether the emerged hall call coincides with the cage calls of the self cage
AT_d : Sum of the incremental predicted arrival time of the already assigned hall calls to the other cage
AET_d: Maximum of the predicted arrival time plus elapsed time of the already registered hall calls since their assignment to the other cage
DNP_d: Difference of the number of passengers between the self and other cage
DNC_d: Difference of the number of assigned hall calls between the self and other cage
CCS_d : Check the coincident service
CSR_d : Check the separate riding for identical destination

3.2.2 Assigning Algorithm

In the GNP with ACO controller, firstly, the information on the elevator system is transferred to the System Information Judgment Part. In this part, the new hall call is classified based on the following three terms, the degree of the variance of the elevator positions VP_{sd}, the origin floor and direction of the new hall call EF_{sd} and the destination floor of the new hall call DF_{sd}.

Secondly, a candidate cage with the minimum value of the evaluation function is selected in the Cage Selection Part. A candidate cage is selected by the following equation. First, the cage evaluation function $e(i)$ of cage i is calculated by Eq.(7).

$$e(i) = \sum_{p \in P} w_p \cdot x_p(i), \quad (7)$$

where,
P : set of suffixes of nodes transited in the cage selection part (P is determined by node transition)
w_p: weight of the cage selection node p (w_p is optimized during evolutionary process)
$x_p(i)$: normalized value of evaluation item X of cage i at the cage selection node p
The normalized value $x_p(i)$ is calculated by Eq. (8)

$$x_p(i) = \frac{X_p(i)}{X_{AveMax}} \quad , \quad (8)$$

where,
$X_p(i)$: value of evaluation item X of cage i at the cage selection node p
X_{AveMax}: maximum value of averaged evaluation item X over past 5 minutes among cages

The reason of using the normalized value of $x_p(i)$ is that different evaluation items have different scales. As for the evaluation item $\{CHC_{sd}, CCS_d\}$, $x_p(i) = 0$ if satisfied, and $x_p(i) = 1$ if not satisfied. It is reversed in the case of $\{CSR_d\}$. Finally, the candidate cage d is selected by Eq.(9)

$$d = arg \min_{i \in I} e(i) \quad , \quad (9)$$

where, I: set of cage IDs

Then, the selected candidate cage d is evaluated again by individual evaluation items one by one to confirm whether it is the optimal one or not in the Cage Judgment Part. In cage judgment nodes in this part, the binary judgment like Eq.(10) is carried out except for $\{CHC_sd, CCS_d, CSR_d\}$.

$$y_j(d) \leq r_j^Y \qquad j \in J, \quad (10)$$

where,
J : set of suffixes of nodes in the cage judgment part
$y_j(d)$: normalized value of evaluation item Y of cage d at the cage judgment node j
r_j^Y : threshold parameter of evaluation item Y of the cage judgment node j (r_j^Y is optimized during evolutionary process)
$y_j(d)$ is also calculated by the following equation similar to Eq.(8).

$$y_j(d) = \frac{Y_j(d)}{Y_{AveMax}}, \quad (11)$$

$Y_j(d)$: value of evaluation item Y of cage d at the cage judgment node j
Y_{AveMax}: maximum value of averaged evaluation item Y over past 5 minutes among cages

As for $\{ CHC_{sd}, CCS_d, CSR_d \}$, the binary judgment (satisfy/not) is done. If Eq.(10) is satisfied and cage judge-

Table 1: Specifications of Elevator Simulator

Items	Value
Number of Floors	16
Number of Shafts(Cages)	6(12)
Floor Distance $[m]$	4.5
Max. Velocity $[m/s]$	2.5
Max. Acceleration $[m/s^2]$	0.7
Jerk $[m/s^3]$	0.7
Cage Capacity $[person]$	20
Time for Opening Door $[s]$	2.0
Time for Closing Door $[s]$	2.3
Time for Riding $[s/person]$	1.0
Passenger Density $[person/h]$	3000

ment node j is connected to the node in the Hall Call Assignment Part, then the new hall call is assigned to the optimal cage d in the Hall Call Assignment Part. Otherwise, i.e., the candidate cage d does not satisfy Eq.(10), which means the condition of evaluation item Y is not satisfied, then, the node transition is resumed from the cage selection part in order to select another candidate cage again.

Finally, in the Hall Call Assignment Part, the new call is assigned to the candidate cage by cage assignment nodes. Node transition returns to the system information judgment part after assignment, and the same procedures are executed for the next call.

3.2.3 Fitness Function

The fitness function of GNP individual is calculated by a weighed sum of the waiting time, maximum waiting time, one cage service and loops of GNP as follows.

$$Fitness = \frac{1}{N}\sum_{n=1}^{N} {t_n}^2 + w_t\cdot(t_{max})^2 + w_n\cdot(n_c)^2 + w_l\cdot n_l^2, \quad (12)$$

where,
N : total number of passengers
t_n : waiting time of the nth passenger
t_{max} : maximum waiting time among N passengers
n_c : total number of passengers experiencing one cage service
n_l : number of loops of GNP per an hour evaluation
w_t, w_n, w_l: weighting coefficient

The number of loops of GNP transition is considered in the fitness because it deteriorates the performances of GNP.

All terms in this function are expected to minimize due to its definitions described above. Thus, an individual with smaller fitness value means that it has a better structure and fitter parameters.

4. SIMULATIONS

4.1 Simulation Conditions

In this paper, we have studied the effectiveness of the proposed GNP with ACO in a typical office building, which has 16 floors and 6 double-deck elevators running at the speed of 2.5m/s. Table 1 shows the specifications of the system simulator. Simulations are executed under 5 kinds of random sequences considering the probabilistic feature of DDES.

Table 2: Evolutional Conditions of GNP with ACO

Items	Value
Node Size	91+Initial Boot Node
Crossover Rate P_c	0.1
Mutation Rate P_m	0.01
Evaluation Time $[h]$	2
Special Generation	every 10 generations
Evaporation Rate ρ	0.2
w_t, w_n, w_l	0.007, 0.003, 0.6

4.2 Results and Discussions

In this section, we show the performances of the proposed algorithm comparing to the simple GNP.

4.2.1 Experiment1

We simulate EGSCS under different population sizes of 30, 100 and 200 using the proposed method (GNP with ACO(M), GNP with ACO(MC)) and a conventional method (GNP without ACO). GNP with ACO(M) means to carry out only mutation using pheromone information at special generations, while GNP with ACO(MC) means to carry out both mutation and crossover using pheromone information at special generations. The parameters of the proposed GNP with ACO are set as shown in Table 2.

The fitness curves of the proposed method are compared with a conventional method in Figure.5. Compared to GNP without ACO, the proposed method GNP with ACO(M), GNP with ACO(MC) converges faster than GNP without ACO. And it is clear from the figure that the fitness values of GNP with ACO(M) and GNP with ACO(MC) are better than GNP without ACO, and it is also clear that the fitness value of GNP with ACO(MC) is better than GNP with ACO(M) in the case of the population size of 30. With the increase of the population size, GNP with ACO(MC) and GNP with ACO(M) get almost the same fitness value. However, GNP with ACO(MC) has a faster convergence speed than GNP with ACO(M).

To sum up, the proposed algorithm converges to a certain value at an early generation. And the fitness values of the proposed method are better than the conventional GNP without ACO.

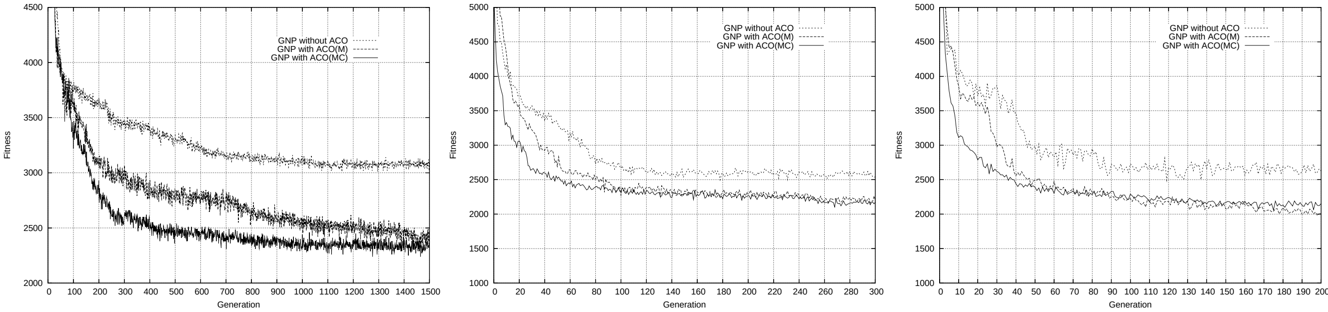
4.2.2 Experiment2

Simulations were implemented using different numbers of special generations under regular time. We set special generations every 5, 10 and 20 generations using different population sizes(30, 70 and 200).

The fitness curves are shown in Figure.6. The more special generations are set in the evolutionary process, the faster convergence speed is obtained in all of the population sizes. But, too many special generations or too few special generations harm the evolution, i.e., too much exploitation and too little exploitation is not a good way for the evolution. The good trad-off between exploitation and exploration can be realized by changing the number of special generations. This experiment shows that setting special generations every 10 or 20 generations is appropriate for EGSCS.

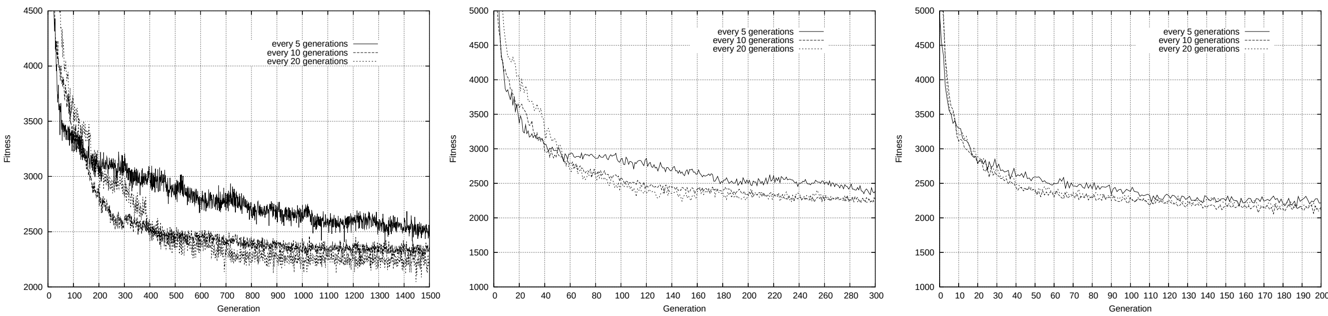
4.2.3 Experiment3

In this subsection, we examined the performances of the proposed method. The performance of the average waiting



(a)Population Size 30 (b)Population Size 100 (c)Population Size 200

Figure 5: Fitness curves of the proposed method in different population size .



(a)Population Size 30 (b)Population Size 70 (c)Population Size 200

Figure 6: Fitness curves when setting special generations every 5, 10 and 20 generations.

Table 3: Performances of AWT, LWP and NC of the Proposed Method.

	GNP without ACO	GNP with ACO(MC)
AWT(s)	32.79	31.67
LWP(%)	12.93	11.88
NC(person)	792	681

time (AWT), the percentage of passengers waiting more than 60s (LWP) and total number of passengers experiencing one cage service(NC) are shown in Table 3. AWT is reduced by 3% in the regular time. In addition, LWP and NC are reduced by 7% at least, a significant improvement.

5. CONCLUSIONS

This paper has proposed GNP with ACO using pheromone information as an enhanced evolutionary method of GNP. The proposed algorithm can save much time in searching the solution space than the conventional method. The performances of it are also better than the GNP without ACO. When the proposed method is applied to the elevator group supervisory control system, the proposed method provided convenient and comfortable services for passengers. As a result, it is found that the proposed algorithm makes a good trade-off between exploitation and exploration.

6. ADDITIONAL AUTHORS

Additional authors: Sandor Markon (Fujitec Co. Ltd, email: `markon@rd.fujitec.co.jp`).

7. REFERENCES

[1] S. Mabu, K. Hirasawa and J. Hu, "A Graph-Based Evolutionary Algorithm: Genetic Network Programming (GNP) and Its Extension Using Reinforcement Learning", *Evolutionary Computation, MIT Press*, Vol. 15, No. 3, pp. 369-398, 2007.

[2] M. Dorigo, V. Maniezzo and A. Colorni, "Ant System: Optimization by a Colony of Cooperating Agents", In *IEEE Transactions on Systems, Man and Cybernetics, Part-B*, Vol. 126, No. 1, pp. 29-41, 1996.

[3] M. Dorigo and L. M. Gambardella, "Ant colony system: a cooperatice learning approach to the traveling salesman problem", In *IEEE Transactions on Evolutionary Computation*, Vol. 1, No. 1, pp. 53-66, 1997.

[4] Y.F. Dong; J.H. Gu; N.N. Li; X.D. Hou and W.L. Yan, "Combination of Genetic Algorithm and Ant Colony Algorithm for Distribution Network Planning " In *Proc. of Machine Learning and Cybernetics*, pp. 999-1002, August 2007.

[5] K. Hirasawa, T. Eguchi, J.Zhou, L. Yu, J. Hu and S. Markon, "A Double-deck Elevator Group Supervisory Control System using Genetic Network Programming", *IEEE Transactions on Systems, Man and Cyvernetics, Part-C*, (to appear).

[6] G. Barney and S. dos Santos, *Elevator Traffic Analysis, Design and Control, Second Ed*, Peter Peregrinus Ltd, 1985.

[7] J. W. Fortune, "Predestination Hall Call Selection for Double-deck Lifts (3D Encoding)", In *, Elevator World*, pp. 126-133, August 2005.