Effects of Passenger's Arrival Distribution to Double-deck Elevator Group Supervisory Control Systems using Genetic Network Programming

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

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

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

Sandor Markon
Fujitec Co. Ltd
Big Wing, Hikone, Shiga
522-8588, Japan
markon@rd.fujitec.co.jp

ABSTRACT

The Elevator Group Supervisory Control Systems (EGSCS) are the control systems that systematically manage three or more elevators in order to efficiently transport the passengers in buildings. Double-deck elevators, where two cages are connected with each other, are expected to be the next generation elevator systems. Meanwhile, Destination Floor Guidance Systems (DFGS) are also expected in Double-Deck Elevator Systems (DDES). With these, the passengers could be served at two consecutive floors and could input their destinations at elevator halls instead of conventional systems without DFGS. Such systems become more complex than the traditional systems and require new control methods Genetic Network Programming (GNP), a graph-based evolutionary method, has been applied to EGSCS and its advantages are shown in some previous papers. GNP can obtain the strategy of a new hall call assignment to the optimal elevator because it performs crossover and mutation operations to judgment nodes and processing nodes. In studies so far, the passenger's arrival has been assumed to take Exponential distribution for many years. In this paper, we have applied Erlang distribution and Binomial distribution in order to study how the passenger's arrival distribution affects EGSCS. We have found that the passenger's arrival distribution has great influence on EGSCS. It has been also clarified that GNP makes good performances under different conditions.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.
GECOO'07, July 7–11, 2007, London, England, United Kingdom.
Copyright 2007 ACM 978-1-59593-697-4/07/0007 ...$5.00.

Categories and Subject Descriptors

I.2.8 [**Artificial Intelligence**]: Problem Solving, Control Methods, and Search—*Graph and tree search strategies*; G.3 [**Probability and Statistics**]: Distribution functions, Queueing theory

General Terms

Performance

Keywords

Elevator Group Supervisory Control System, Passengers' arrival, Erlang distribution, Genetic Network Programming

1. INTRODUCTION

There have been installed more and more high-rise buildings in the cities for the spatial and economical considerations. To provide transportation services among floors in the building, elevator systems are installed as primary service facilities. Elevator group supervisory control systems (EGSCS)[1] are responsible for controlling elevators to provide convenient and comfortable services for passengers. The new generation elevators, Double-deck elevators[2] are designed to connect two cages in an elevator shaft. This allows passengers on two consecutive floors to use the elevator simultaneously, significantly increasing the transportation capacity of elevator systems. Meanwhile, with the Destination Floor Guidance System (DFGS)[3, 4], EGSCS could use the information of the destination floor before passengers get in elevators, instead of waiting until the cage call button is operated in elevators.

Furthermore, since the elevator system is driven by passenger's arrival, its probability distribution characterizes the traffic in the elevator systems. In order to get tractable analytical results, researchers have used Exponential distribution to model the elevator systems for many years. However, Erlang distribution is a more general distribution of nonnegative random variables. Especially, it could be used to model passengers who got off at the railway stations adjacent to the buildings arrive at the floors of elevators. In this paper, we use Erlang distribution and Binomial distribution to build more general passenger's arrival model and to study how the passenger's arrival distribution affects EGSCS using DDES with DFGS and Genetic Network Programming (GNP)[5, 6], which is a newly developed graph-based evolutionary computation method.

The paper is organized as follows. Section 2, 3 and 4 gives an overview of DDES, GNP and describes the details of Erlang and Binominal distribution used for EGSCS, respectively. Section 5 explains EGSCS using GNP. Section 6 shows the simulation conditions and results. Finally, some conclusions are devoted in section 7.

2. ELEVATOR GROUP SUPERVISORY CONTROL SYSTEMS (EGSCS)

Elevator Group Supervisory Control Systems (EGSCS) are control systems that manage multiple elevators in a building in order to efficiently transport the passengers. The systems assign a service car for a new passenger waiting in a hall. The assignment is a kind of real-time scheduling problem for transportation systems. The performance of EGSCS is measured by several criteria such as the average waiting time of passengers, the percentage of passenger's waiting more than 60 seconds, and power consumption [7], EGSCS manages elevators to minimize the evaluation criteria; it is, however, difficult to satisfy all criteria at the same time. In this paper, the following two criteria are used.

1) **Average waiting time (AWT)** is the average time until the service elevator arrives at the floor after a passenger presses a hall call button.

2) **Average traveling time (ATT)** is the average time excluding AWT after the passengers get into the cage until drop off at the destination floor.

The passenger traffic pattern in modern buildings with EGSCS varies considerably throughout a typical business day. Early in the morning, most of the passengers travel from the lobby to the upper floors (Up-peak), while at the end of the day, most of the passengers leave the floors and travel primarily to the lobby in order to exit the building (Down-peak). And other part of the day has its own characteristic patterns (Regular). Different traffic patterns have very different effects, and each pattern requires its own analysis. Up-peak and down-peak elevator traffic are not simply equivalent in the sense of opposite directions, as one might initially guess. Down-peak traffic has many arrival floors and a single destination, while up-peak traffic has a single arrival floor and many destinations. So, we study the systems with three types of traffic patterns, i.e., "Up-peak Time", "Down-peak Time" and "Regular Time".

2.1 Destination Floor Guidance System(DFGS)

In traditional elevator systems, EGSCS consists of hall call buttons and car call buttons. If a passenger wants to go to another floor, he presses a direction (hall call) button and waits for an elevator to arrive, then enters the elevator and presses a destination floor (car call) button in the elevator. The EGSCS selects a suitable elevator when a passenger presses the hall call button. In order to obtain more accurate information on passenger's destination, Destination Floor Guidance System (DFGS) has been developed, so that passengers can input their destinations at elevator halls. At each floor there is a keypad where the passenger selects which floor they wish to go to. The system then guides the passenger to an elevator that will be stopping at their destination floor. There are no floor buttons inside the cage. Such systems claim that average waiting time can be reduced by up to 30%, by grouping passengers with common destinations into the same cage, and thus reducing the number of stops that need to be made.

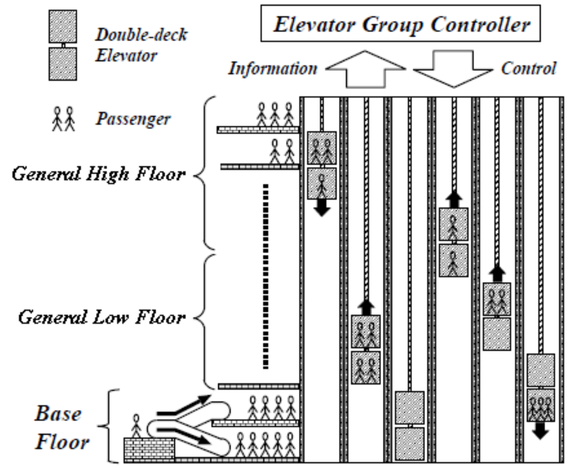


Figure 1: Outline of Double-Deck Elevator System.

2.2 Double-Deck Elevator System (DDES)

Recently, for improving the capability of EGSCS, the Double-Deck elevator, where two cages are connected with each other, is expected as the next generation elevator. It allows the passengers at two consecutive floors could be serviced simultaneously. Such a scheme could be efficient in buildings where the traffic would have a stopping at every floor. Architecturally, double-deck elevators occupy less building core space than traditional single-deck elevators do for the same level of traffic. This allows much more efficient use of space, as the floor area required by elevators tends to be quite significant.

Figure 1 shows the outline of Double-Deck Elevator Systems (DDES). In DDES, a passenger can in principle board either the lower or upper cage. Here, instead of "upper cage" and "lower cage", we also use the terms "self cage" and "other cage" in a more general sense. As the upper cage could not get to the bottom floor, we divide the base floor into "up-base floor" and "down-base floor". The two bottom floors are named "Base Floor". With the DFGS, when passengers come to the lobby of the building, the panel would tell them that they should go to the up-base floor or down-base floor to be serviced. Obviously, Double-Deck Elevator Systems (DDES) become more complex in their behavior than conventional Single-Deck Elevator Systems (SDES).

DDES has specific features as shown below, and their careful consideration is expected to improve the performances of group supervisory control.

One Cage Service: Self cage stops without any service while the other cage serves passengers at the floor. This situation causes not only the deterioration of transportation capability but also psychological stress to passengers.

Coincident Service: Both cages serve passengers at a stop. Coincident service can contribute to improve both transportation capability and comfortable riding.

Separate Riding for Identical Destination: Passengers for the identical destination ride on both cages. Therefore, the transportation capability deteriorates by two stops at the same floor.

3. PASSENGER'S ARRIVAL DISTRIBUTION

Queueing systems use a particular form of state equations known as Markov chains[8]. To drive a queueing model that represents real systems, we prefer a form that is simple or tractable, but require that it is sufficiently realistic. It has been found convenient to work with probability distribution which exhibits the memoryless property, as it commonly simplifies the mathematics involved. As a result, queuing models are frequently modeled as Poisson Processes through the use of the exponential distribution.

However, using this distribution, arrivals occur completely random in time. Sometimes we need control the distribution in order for it to be close to the real one. Especially, if the passengers who got off at the stations adjacent to buildings arrive at the floors of elevators by the batch, we should consider the above. So here, we use the Erlang distribution to realize the time interval of passenger's arrival and use the Binomial distribution to assign the number of the batch.

3.1 Erlang Distribution

A random variable X has an Erlang-k ($k = 1, 2, \ldots$) distribution with mean k/μ if X is the sum of k independent random variables $X_1, \ldots, X_k$ having a common exponential distribution with mean $1/\mu$. The common notation is $E_k(\mu)$ or briefly E_k. The density of $E_k(\mu)$ distribution is given by

$$f(t) = \mu \frac{(\mu t)^{k-1}}{(k-1)!} e^{-\mu t}, t > 0. \tag{1}$$

The distribution function equals

$$F(t) = 1 - \sum_{j=0}^{k-1} \frac{(\mu t)^j}{j!} e^{-\mu t}, t \geq 0. \tag{2}$$

The parameter μ is called the scale parameter, k is the shape parameter. In Figure 2 we display the density of the Erlang-k distribution with mean 1 (so $k/\mu = 1$) for various values of k.

When $k = 1$, the distribution is reduced to the exponential distribution which realizes the random process. As the shape parameter k increases, the distribution becomes like symmetry. When $k \geq 30$, it closes to normal distribution. When $k \to \infty$, it becomes a delta function at the value of k/μ. So, the Erlang-k distribution could make the time interval of passenger's arrival to be the form from completely random to a certain value. It provides wider applicability to real systems. The Figure 3 illustrates that Erlang arrivals spread out much more equally over time than exponential arrivals.

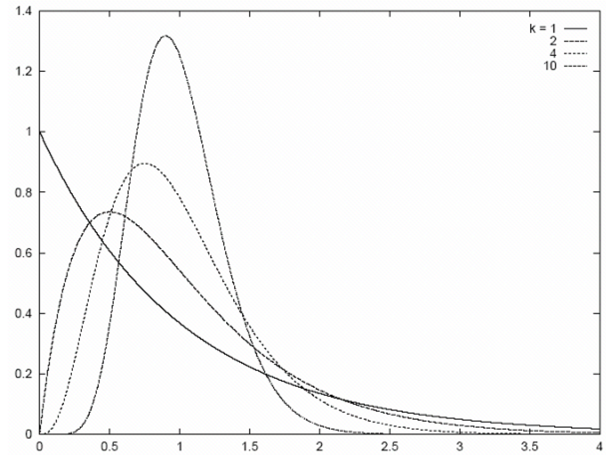


Figure 2: The density of Erlang-k distribution.

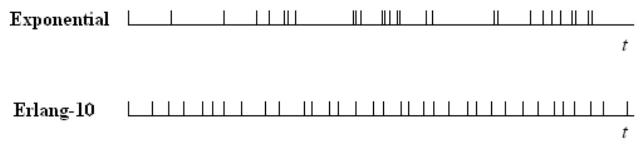


Figure 3: Exponential arrivals and Erlang-10 arrivals.

3.2 Binomial Distribution

The binomial distribution P_k is a discrete probability distribution taking from 0 to n with mean of np. It denotes the number of successes in a sequence of n independent yes/no experiments, each of which yields success with probability p

$$P_k = C_k^n p^k (1-p)^{n-k}, k = 0, 1, 2, \ldots, n. \tag{3}$$

Where,

$$C_k^n = \frac{n!}{k!(n-k)!}, 0 \leq p \leq 1. \tag{4}$$

We can realize passenger's arrivals in the batch mode having the mean arrival rate of $\frac{\mu n p}{k}$ by combining $E_k(\mu)$ and P_k.

4. GENETIC NETWORK PROGRAMMING (GNP)

Figure 4 shows the basic structure of GNP. As an extension of GA and GP, GNP has been proposed to have a network structure where functional nodes are connected by directed branches. GNP program is composed of one start node and plural judgment nodes and processing nodes. The start node has no functions and no conditional branches. Judgment nodes have decision functions with conditional branches. Each judgment node returns a judgment result and determines the next node to be executed. Processing nodes work as action functions. After the start node, the current node is transferred according to the node connections and judgment results. In processing nodes, actions are

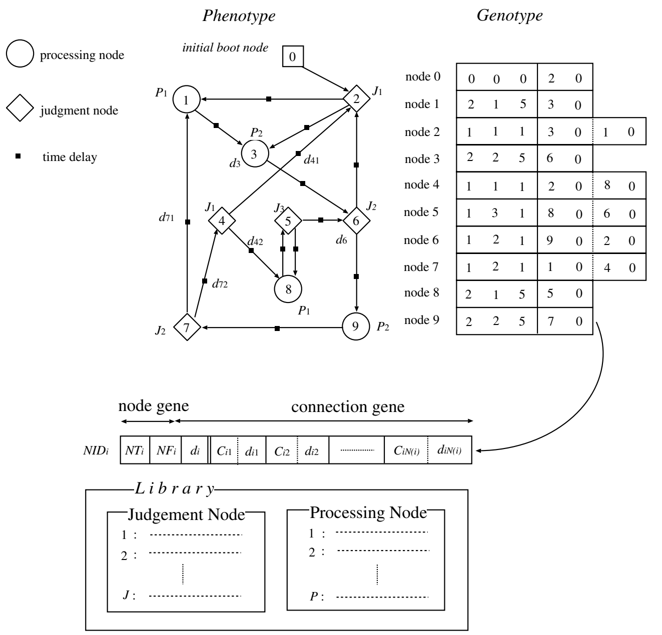


Figure 4: Basic structure of GNP.

conducted to environments. All kinds of judgment and processing function labels (Judgment node: $\{1, 2, \ldots, J\}$, Processing node:$\{1, 2, \ldots, P\}$) are set up in the libraries, which are prepared by the designers. The node transition begins from a start node, and there is no terminal node.

GNP has two kinds of time delays: one spends on judgment nodes or processing nodes, and the other one spends on node transition. Since time delays are listed in each node gene and are unique attributes of each node, GNP can evolve flexible programs considering time delays.

As shown in Figure 4, GNP can be illustrated by its "Phenotype" and "Genotype". Phenotype GNP shows the directed graph structure where nodes are connected by directed branches, and Genotype GNP provides the chromosomes encoded into bit-strings. The structure of the gene of node i is set as shown in Figure 4. There are node genes and connection genes in the genes of nodes. NT_i is the allele of node type (0: start node, 1: judgment node, 2: processing node). NF_i indicates the function label which is defined in the library. d_i is the time delay of node i. C_{ik} denotes the k^{th} connecting node number from the current node i and d_{ik} shows the time delay of the transition.

In evolutionary computation, each individual is evaluated in the problem environment. Then the offspring who can survive at the next generation is decided by fitness. Crossover, Mutation, Tournament Selection and Elite Preservation are used as the genetic operators of GNP. The outline of evolution is described as follows:

1. Generate initial population and calculate the fitness of initial population;
2. Execute tournament selection, genetic operations to individuals and generate new individuals for the next generation;
3. Calculate the fitness of the new individuals;
4. Repeat 2-3 until the terminal condition meets.

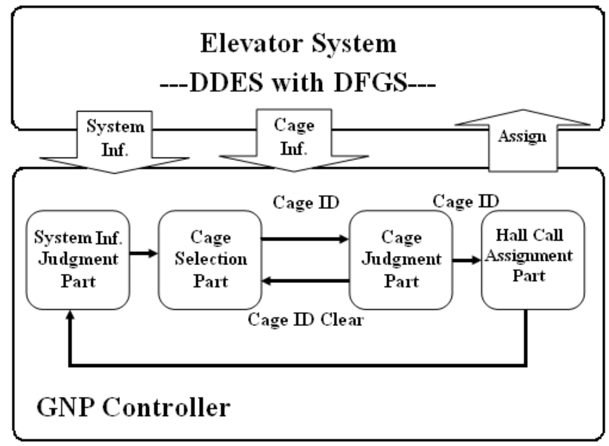


Figure 5: Structure of the proposed system.

5. APPLICATION OF GNP TO EGSCS

Double-Deck Elevator Systems with Destination Floor Guidance Systems are so complex in that the assignment of the optimal cage to each new hall call is fairly difficult due to the enormous amount of information obtained. GNP is expected to be appropriate for the assignment problem in elevator systems. The reason is that: GNP can realize a rule based Elevator Group Supervisory Control System (EGSCS) due to its directed graph structure with judgment nodes and processing nodes, which makes EGSCS more flexible in different traffics. And also, EGSCS can be generated by an evolutionary method with mutation, crossover and selection, which could develop new efficient and effective rules that elevator experts can not imagine as well as saving the time for designing EGSCS.

The structure of Double-Deck Elevator System (DDES) with Destination Floor Guidance System (DFGS) using GNP is shown in Figure 5. The Elevator Group Supervisory Control System (EGSCS) includes Elevator System and GNP controller. When a hall call occurs, System Information and Cage Information are collected. Then the GNP Controller uses these data, and does some calculation and evaluation. The GNP Controller consists of the System Information Judgment Part, Cage Selection Part, Cage Judgment Part and Hall Call Assigning Part. The information is transferred through those parts.

5.1 Evaluation Items

In our proposed method the following 12 evaluation items are defined and employed 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 arrival time plus elapsed time since the assignment of the hall calls 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 (passenger number/ 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 car 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 arrival time plus elapsed time since the assignment of the hall calls 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

5.2 Assigning Algorithm

In the GNP controller part, firstly, the information on the elevator system is transferred to the System Information Judgment Part. There, the degree of variance of the elevator position, the floor and direction of the new hall call and the destination floor of the new hall call are judged by the system information judgment nodes. An activated node in the system information judgment part is transferred to an appropriate node in the cage selection part. In other words, the best initial node in the cage selection part is determined by studying the elevator positions and the information on the emerged and destination floors of a new hall call.

Secondly, a candidate cage with the minimum value of the evaluation function is selected in the Cage Selection Part. The evaluation function might include some of the 12 items, which are mentioned before (AT_{sd}, AET_{sd}, NP_{sd}, NC_{sd}, RR_{sd}, CHC_{sd}, AT_d, AET_d, DNP_d, DNC_d, CCS_d, CSR_d). The candidate cage should be the self cage, not the other cage, because we suppose that a new hall call is assigned to the best cage in the self cages. The self cage could be the upper cage or the lower cage. In the cage selection part, we have only processing nodes and each of these continues to calculate one of the 12 items of all self cages by node transition until an activated node in the cage selection part moves to the node in the cage judgment part. Different from the conventional methods, the number and combination of the items for the evaluation function are decided by evolution. In other words, the connection from the current processing node to the next node, which might be another node in the cage selection part or the node of the cage judgment part, is determined by evolution of GNP. Meanwhile, each item in the evaluation function has a weight adjusted by GA. The cage evaluation function $e(i)$ of cage i is calculated by Eq.(5). Now we suppose that item X for cages is evaluated at the cage selection node $p \in P$

$$e(i) = \sum_{p \in P} w_p \cdot x_p(i). \tag{5}$$

The normalized value $x_p(i)$ of the evaluation item X of cage i at the cage selection node p is calculated by Eq. (6)

$$x_p(i) = \frac{X_p(i)}{X_{AveMax}} \quad , \tag{6}$$

where,
P: Set of suffixes of nodes transited in the cage selection part
w_p: Weighting parameter of cage selection node p
$X_p(i)$: Value of evaluation item X of cage i at the cage selection node p
$x_p(i)$: Normalized 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

As for the evaluation item $\{CHC_{sd}, CCS_d\}$, $x_p(i) = 0$ if the condition is satisfied, and $x_p(i) = 1$ if not satisfied. As for the evaluation item $\{CSR_d\}$, it is vice versa. Finally, the candidate cage d is selected by Eq, (7)

$$d = arg \min_{i \in I} e(i) \quad , \tag{7}$$

where,
I: set of cage IDs

Then, the selected candidate cage d is evaluated again by individual evaluation items each by each in the Cage Judgment Part in order to study if the candidate cage would be a really satisfactory one in point of each evaluation item. In concrete, the binary judgment like Eq. (8) is carried out except $\{CHC_sd, CCS_d, CSR_d\}$ in cage judgment nodes j. If

$$y_j(d) \leq r_j^Y \qquad (j \in J), \tag{8}$$

where,

$$y_j(d) = \frac{Y_j(d)}{Y_{AveMax}} \tag{9}$$

J: Set of suffixes of nodes in the Cage Judgment Part
r_j^Y :parameter of evaluation item Y at cage judgment node j
$Y_j(d)$: Value of evaluation item Y of cage d at the cage judgment node j
$y_j(d)$: normalized value of 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 the candidate cage d satisfies Eq(8), then the new hall call is assigned to the best cage d in the Hall Call Assignment Part. Otherwise, the node transition is resumed from the cage selection part in order to select another candidate cage.

After the hall call assignment to the best cage completes, GNP stops transitions until another new hall call occurs and the transition begins at the node in the System Information Part, which is connected from the node in the Hall Call Assignment Part.

It should be noted that all the connections in the nodes in four parts of GNP mentioned above could be changed by evolution.

5.3 Node Functions

There are 4 kinds of nodes in the parts of the algorithm described in the previous sub-section. They are as follows.

$< System\ Information\ Judgment\ Node >$
$J^{VP_{sd}}$: Judge the degree of variance of the elevator position
$J^{EF_{sd}}$: Judge the floor and direction of the new hall call
$J^{DF_{sd}}$: Judge the destination floor of the new hall call

$< Cage\ Selection\ Node >$
$S(X)$: select evaluation item X from 12 items by the node transition in the Cage Selection Part and calculate Eq. (7)

$X \in \{AT_{sd}$, AET_{sd}, NP_{sd}, NC_{sd}, RR_{sd}, CHC_{sd}, AT_d, AET_d, DNP_d, DNC_d, CCS_d, $CSR_d\}$

$< Cage\ Judgment\ Node >$

$J^Y(d)$: Judge whether $y_j(d) \leq r_j^Y$ is satisfied or not

$Y \in \{AT_{sd}, AET_{sd}, NP_{sd}, NC_{sd}, RR_{sd}, AT_d, AET_d, DNP_d, DNC_d\}$

$J^Z(d)$: Judge whether Z of cage d is satisfied or not

$Z \in \{CHC_{sd}, CCS_d, CSR_d\}$

$< Hall\ Assignment\ Node >$

$A(d)$: Assign the new hall call to cage d

5.4 Fitness Function

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

$$Fitness = \frac{1}{N}\sum_{n=1}^{N} {t_n}^2 + w_t \cdot (t_{max})^2 + w_c \cdot (N_c)^2 + w_l \cdot l^2 \quad (10)$$

N: Total number of passengers
t_n: Waiting time of n-th passenger
t_{max}: Maximum waiting time among N passengers
N_c:Total number of passengers experiencing one cage service
l: Number of loops of GNP per an hour evaluation
w_t, w_c, w_l: Weighting coefficient

6. SIMULATIONS

6.1 Simulation Conditions

In this paper, we study how the passenger's arrival distribution affects EGSCS using GNP in a typical office building, having 16 floors and 6 double-deck elevators running at 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.

As shown in Table 2, simulations are implemented for the three kinds of traffic flow, "Regular Time", "Up-peak Time" and "Down-peak Time". The row of the Table represents the floor where passengers emerge, and the column represents the floor where passengers plan to go. Here, passengers emerge at a floor whose type is listed as "Base Floor (Base)" or "General Floor (Gen.)".

The parameters of the method using GNP are set as shown in Table 3.

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 $[person/h]$	1.0
Passenger Density $[person/h]$	
—Regular Time	3000
—Up-peak	2700
—Down-peak	3300

Table 2: Traffic Flow Ratios.

	Regular		Up-peak		Down-peak	
	Base	Gen.	Base	Gen.	Base	Gen.
Base	—	5	—	19	—	2
General	5	1	1	2	19	1

6.2 Results and Discussions

The fitness curves of the best GNP individual in each traffic using Erlang distribution are shown in Fig.6 ((a) Regular Time, (b) Up-peak Time, (c) Down-peak Time). The fitness curve of the best individual is the average over the results obtained from 5 kinds of random sequences. The shape of the fitness curves differs traffic flows by traffic flows. We made simulations under the different parameters of $k = 1, 10$ and 50 of Erlang distribution. When k is large enough, the time interval of the passenger's arrival is regular. From the figures of fitness curves, we could find two important things when parameter k increases. Firstly, the speed of the evolution in each traffic becomes faster. In other words, the case of using small k needs more generations for convergence. Secondly, the lager value of k would make the lower fitness value.

Also, we made the simulations with the batch arrival using Erlang and Binomial distribution. In this case, the passengers emerge at floors by batch. Here, we set the parameters of Binomial distribution as $n = 50, p = 0.06$. It means the mean of the distribution is $n \times p = 3$. It indicates 3 passengers on average might emerge at the same time. Figure 7 shows the fitness curves of each traffic using Erlang distribution in the batch mode. The batch passengers make the system more complex, so we could see the fitness is higher than the cases without batch, but it is still under the influence of the parameter of Erlang distribution.

The fitness values of down-peak traffic are lower than those in up-peak and regular traffic because down-peak traffic has more diversified flows of traffic than up-peak time, and also it is simpler, i.e., mainly one direction, than regular traffic.

The performance of AWT(average waiting time) and ATT (average traveling time) are shown in Table 4 ((a) Performances without batch. (b) Performances with batch.). It is shown from Table 4 that most of the values are decreased except for Up-peak if parameter k increases.

Table 3: Evolutional Conditions of GNP.

Items	Value
Generation	300
Population Size	300
—Crossover	120
—Mutation	170
—Elite	10
Node Size	91+Initial Node
Crossover Rate P_c	0.1
Mutation Rate P_m	0.01
Evaluation Time $[h]$	2
w_t	0.007
w_c	0.001
w_l	0.6

Table 4: Average Waiting Time and Traveling Time. (unit: second)

(a) Performances without Batch

	Regular		Up-peak		Down-peak	
	AWT	ATT	AWT	ATT	AWT	ATT
k=1	29.94	63.94	30.06	60.24	27.80	60.31
k=10	28.72	62.52	27.98	58.52	25.29	59.27
k=50	27.67	60.24	26.88	59.28	24.30	57.08

(b) Performances with Batch

	Regular		Up-peak		Down-peak	
	AWT	ATT	AWT	ATT	AWT	ATT
k=1	33.30	68.51	34.20	89.86	34.07	61.44
k=10	34.92	63.99	33.44	90.89	31.11	60.78
k=50	32.88	60.53	32.33	88.18	30.03	59.71

Note that the results in "Up-peak Time" are a bit different. The reason causing those results are due to the inherent properties in up-peak time, where most of passengers emerge at the base floor and go to higher floors[9]. So the performance in this pattern is not influenced by the parameter of Erlang distribution so much.

Figure 6, Figure 7 and Table 4 were obtained using the same value of parameter k when training and testing, while Table 5 shows the average waiting time when different k is used in training and testing. It is found from Table 5 that testing results with k being 1, 10, 50 do not have so much differences when trained using $k = 1$, on the other hand, testing results have much differences when $k = 50$ is used in training, i.e., the larger k is, the smaller AWT is.

Table 5: Average Waiting Time when different k is used in Training and Testing. (unit: second)

(a) Regular time

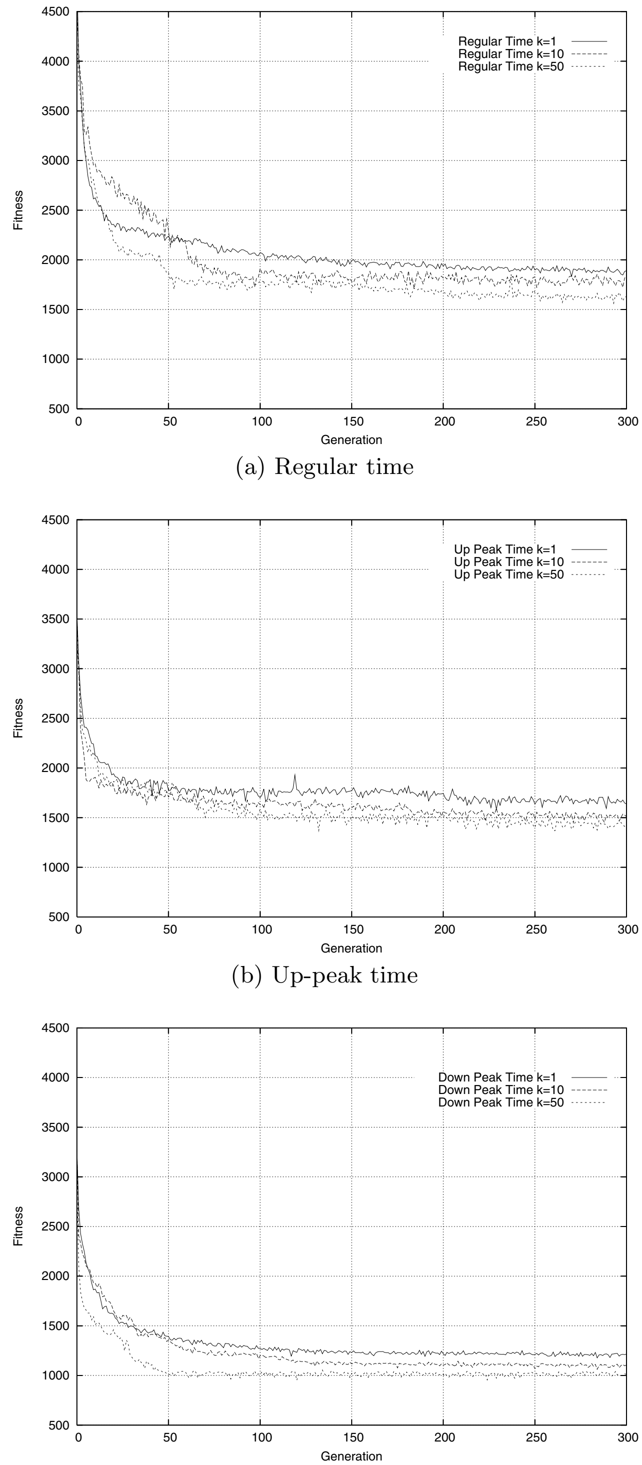
Training \ Testing	k=1	k=10	k=50
k=1	29.94	29.94	30.04
k=10	30.00	28.72	29.12
k=50	29.92	28.96	27.67

(b) Up-peak time

Training \ Testing	k=1	k=10	k=50
k=1	30.06	30.10	28.69
k=10	28.02	27.98	27.85
k=50	30.11	28.02	26.88

(c) Down-peak time

Training \ Testing	k=1	k=10	k=50
k=1	27.80	27.79	26.97
k=10	27.89	25.29	25.30
k=50	27.93	25.29	24.30

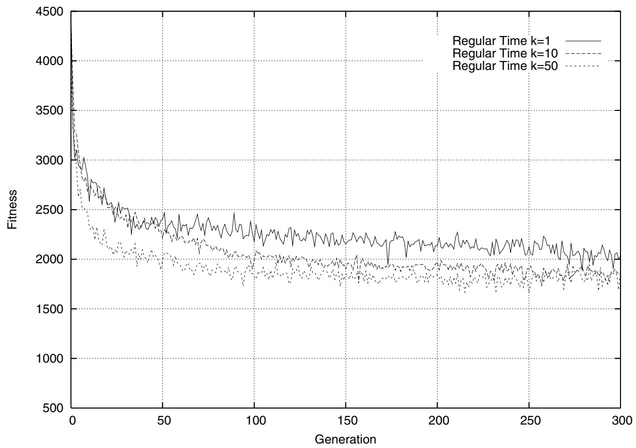


(a) Regular time

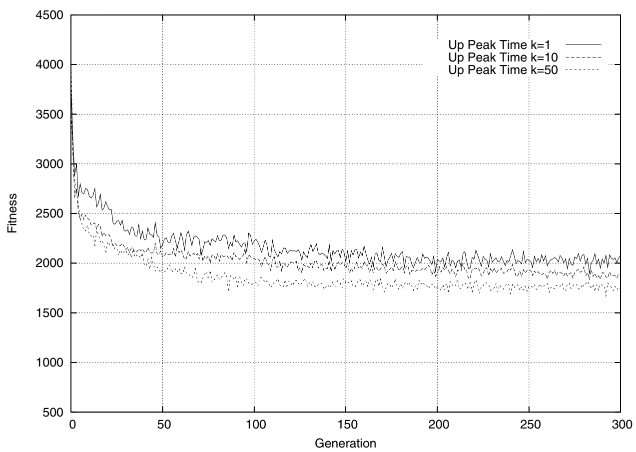
(b) Up-peak time

(c) Down-peak time

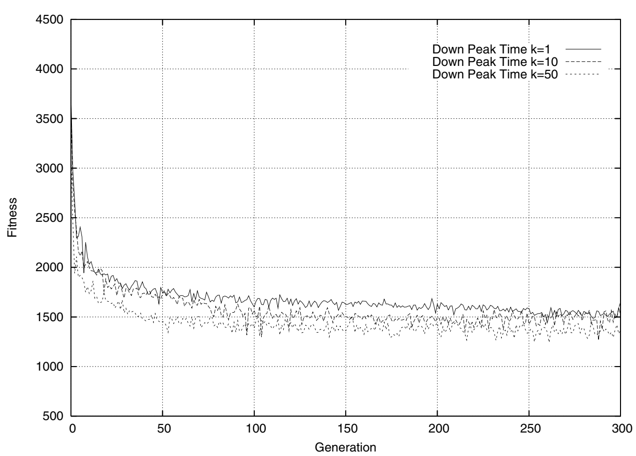
Figure 6: Fitness curves using Erlang distribution.



(a) Regular time



(b) Up-peak time



(c) Down-peak time

Figure 7: Fitness curves using Erlang distribution with batch mode.

7. CONCLUSIONS

In this paper, a method for studying the effects of passenger's arrival distribution to elevator group supervisory control systems using GNP has been proposed. It has been clarified from the simulations of using Erlang and Binominal distribution for passenger's arrival that poisson arrival and batch arrival have great influences on average waiting time and average traveling time in terms of increasing these values.

8. REFERENCES

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

[2] Marja-Liisa Siikonen, "Double-Deck Elevators: Savings in time and space", In *Elevator World*, June 1998.

[3] J. Koehler and D. Ottiger, "An AI-based approach to destination control in elevators", In *AI Magazine*, 2002. 23(3): pp.59–79.

[4] S. Tanaka, Y. Innami and M. Araki, "A study on objective functions for dynamic operation optimization of a single-car elevator system with destination hall call registration", *IEEE international Conference on Systems, Man and Cybernetics*, 2004.

[5] T. Eguchi, K. Hirasawa, J. Hu and N. Ota, "Study of Evolutionary Multiagent Models Based on Symbiosis", *IEEE Trans. on Systems, Man and Cybernetics, Part-B*, Vol. 35, No. 1, pp. 179–193, 2006.

[6] 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 (to appear).*

[7] R. D. Peters, "The theory and practice of general analysis of lift calculations"*, in Elevcon Proc.*, 1992, pp. 197-206.

[8] H.C, Tijms, "Algorithmic Analysis of Queues", Chapter 9 in A First Course in Stochastic Models, Wiley, Chichester, 2003.

[9] D. Pepyne and C. Cassandras, "Optimal dispatching control for elevator systems during uppeak traffic", IEEE Trans. Contr. Syst. Technol., Vol.5, No.6, pp. 629-643, Nov. 1997.