Systems and Computers in Japan, Vol. 34, No. 1, 2003
Translated from Denshi Joho Tsushin Gakkai Ronbunshi, Vol. J84-D-I, No. 2, February 2001, pp. 191–202

Elevator Group Control System Using Multiagent System

Yasuhiro Ogoshi,[1] Haruhiko Kimura,[2] Sadaki Hirose,[1] and Nobuyasu Osato[3]

[1]Toyama University, Faculty of Engineering, Toyama, 930-8555 Japan

[2]Kanazawa University, Faculty of Information Systems Technology, Kanazawa, 920-8667 Japan

[3]Osaka Institute of Technology, Faculty of Information Science and Technology, Japan

SUMMARY

In this study, the authors proposed an elevator group control system using multiagent system, and verified the efficiency of the system in areas characterized by concentration of large crowds of passengers on the main floor, for example, during morning rush-hour traffic. Congestion on the main floor in heavy morning rush-hour traffic areas was previously alleviated using specific traffic control systems providing so-called morning rush-hour operation by forced dispatching of multiple elevators to the main floor. However, since forced dispatching on the main floor affects areas other than the main floor and also the subsequent service on the main floor, it makes overall improvement of transportation efficiency a difficult task. There are published reports relating to morning rush-hour operation, but these deal with centralized control based on overall monitoring and sending instruction commands. Thus, the problem is that if this control system stopped working, it caused overall malfunctioning. In addition, in a system based on overall monitoring and generating optimal control plan in accordance with the received information, it is not easy to generate design updates following a change in the system's design or environment. Therefore, in the current study the authors introduced a multiagent system with distributed processing, in which individual elevators make action decisions by independent learning process. The authors proposed an elevator group control system which uses simplified reasoning to provide for flexible response even in the case of a changed environment, such as when a part of the elevator group has stopped functioning. Syst Comp Jpn, 34(1): 45–58, 2003; Published online in Wiley InterScience (www.interscience.wiley.com). DOI 10.1002/scj.1183

Key words: elevator group control; cage dispatching; morning rush-hour operation; multiagent; reinforcement learning.

1. Introduction

Considering that elevators are major transportation means inside of buildings, it is important that the load is efficiently dispatched to them at times of high congestion such as during morning rush hour. However, operation of elevators is different from that of usual transportation means in that they quickly move between locations in a very short time, which makes the task of their efficient control much more complicated.

One method [1] involves predicting overall hall call wait time, defining evaluation function from the average wait time and the number of people who endured a long wait (usually, when the wait time was 60 seconds or more), and then determining the elevator cage dispatching. After this publication appeared, control methods based on the cage dispatching according to such evaluation functions became the mainstream. Several examples of typical techniques using such evaluation functions are the method which involves area control [2] based on distributing elevators around the building and localizing their respective service regions, and another method which involves detecting characteristic vectors, namely: the number of people getting on and off at each floor at each time moment, and the traffic flow such as direction they are moving in, and then using learning theory [3] for selecting control parameters based on representative characteristic vectors.

Later, other studies pursued innovation to those mainstream control methods based on wait time prediction and evaluation function: a study on control techniques using the expert system approach [4], and a study which demonstrated the efficiency of fuzzy theory approach to rush-hour traffic [5].

However, fuzzy rules must be set in advance to suit the given environment and traffic patterns. In addition, membership function parameters applicable to the fuzzy set must also be readjusted in the cases of changed environment, which makes it difficult to update settings.

Therefore, the authors of this study introduced a multiagent system which is a distributed processing control system, and proposed an elevator group control system capable of flexible adjustment to changed environment, in which the load (reasoning complexity) is alleviated by having each individual elevator make independent judgments. We have also conducted a simulation-based performance comparison in the same environment as in the system developed by Hikita and Komaya [5] which was shown to be very efficient in known studies pertaining to rush-hour traffic.

2. Conventional Elevator Group Control Systems

2.1. Tasks relating to rush-hour traffic

In a situation when many passengers appear on the main floor during rush hour in an office building having several elevators, and when the main floor calls have been dispatched to the elevator cages, it is a complicated task to make allocation for repeatedly appearing new passengers. Therefore, the practice is to have on the main floor several elevators waiting even if there are no hall calls—so-called morning rush-hour service.

However, when during the rush hour extra elevators are dispatched to the main floor, not only service on the other floors suffers, but an awkward traffic situation is created when several elevators operate in linked fashion, so that it is not possible to attain transportation efficiency in proportion to the number of elevators.

2.2. Conventional systems and problems

A typical example of a system designed to accurately determine the number of elevators dispatched to the main floor during rush hour is the system of Hikita and Komaya [5]. In this system the following fuzzy rules are set based on several indices such as

- concentration of equipment (on main floor): cage stopping rate
- service conditions (on main floor): cage standing rate, average number of passengers in elevator, number of takeoffs with passengers, number of takeoffs without passengers
- surplus of equipment (on upper floors): cage waiting rate

and then the control aims at increasing or decreasing the number of dispatched elevators. This technique allows efficiently servicing the main floor without degrading the upper floor services.

However, these fuzzy rules must be set in advance to correspond to the assumed environment and traffic pattern. Moreover, considering that membership function parameters in the fuzzy set should be adjusted to the environment, the design must be updated following changes in the environment.

3. Proposed System

In the proposed system, the following functions are assigned as agents to each individual elevator, and the entire elevator group is designed to function as a multiagent system. This group management system aims at service improvement for the entire group, and is also capable of flexibly adjusting to environment changes:

- independent learning of how to respond to hall calls
- using communication for following the built-in rules to avoid competition when elevators compete in responding to hall calls

During the learning stage, the system adds these rules to the rules that in accordance with the morning rush-hour situation dispatch elevators to the main floor, which is an advantageous point of conventional systems, and then the system supplies the combined learning rules in advance to the agents. The details will be described further starting in Section 3.2. The agents will thereafter be referred to as elevators.

3.1. Description of elevator states

3.1.1. Basic information

In an environment of an N-story building ($N \geq 2$) serviced by M ($M \geq 1$) elevators, the elevators states can be described as shown below. The UP and DOWN buttons among the hall call buttons are denoted as △ and ▽, respectively.

- States of the hall call buttons on n-th floor
  - △ button UP_Xn = {ON, OFF} (n = 1, 2, . . . , N – 1)
  - ▽ button DOWN_Xn = {ON, OFF} (n = 2, 3, . . . , N)
- Movement direction of elevator m
  - Dm = {up, no direction, down}(m = 1, 2, . . . , M)
- Current location of elevator m
  - Lm = {1, 2, . . . , N} (m = 1, 2, . . . , M)
- States of the cage call button of the elevator m on the n-th floor
  - IN_Xmn = {ON, OFF} (m = 1, 2, . . . , M), (n = 1, 2, . . . , N).

Considering all combinations of these states, we shall have:

- Combinations of hall call buttons: set of $2^{2(N-1)}$
- Direction and location: set of $(3N-2)^M$
- States of the cage call buttons: set of 2^{NM}

Therefore, the maximum total number of system states will be $2^{2(N-1)} \cdot [(3N-2) \cdot 2^N]^M$. For example, in the environment of an 11-story building serviced by 6 elevators, there will be approximately 6.87×10^{34} sets of states. Since it is impossible to learn all of these states, the number of states to learn is reduced by subdividing the processing for hall call decisions into processing by learning and processing by built-in rules, as will be shown in Section 3.1.3.

In addition to the above information, the system uses the following information.

3.1.2. Applied information

- Time elapsed after pressing a hall call button on n-th floor [seconds]
  - △ button UP_Tn > 0 (n = 1, 2, . . . , N – 1)
  - ▽ button DOWN_Tn > 0 (n = 2, 3, . . . , N)
- Total number of passengers [people] during the last 5 min (estimated by the increased or decreased load in the cage)
  - △ button UP_5Rn > 0 (n = 1, 2, . . . , N – 1)
  - ▽ button DOWN_5Rn > 0 (n = 2, 3, . . . , N)
- Operation state of elevator m
  - Sm = {wait, forward, transport} (m = 1, 2, . . . , M)
- Information on cage dispatching for elevator m
  - Bm = {1, 2, . . . , N} (m = 1, 2, . . . , M)

3.1.3. Information used for hall call decisions

In the proposed system, for hall calls decision processing is subdivided into processing by learning and built-in processing. The information which was described so far in Sections 3.1.1 and 3.1.2 is organized by what type of processing it is used in.

(1) Information used in processing by learning

The following attribute information pertaining to hall call buttons is used in processing by learning, and the hall calls are determined in accordance with the learning rules described in Section 3.2.1.

- Information regarding ON, OFF of hall calls
  - UP_Xn, DOWN_Xn on each floor
- Time elapsed after the hall call button was pressed
  - UP_Tn, DOWN_Tn on each floor
- Number of passengers having made hall calls for last 5 min
  - UP_5Rn, DOWN_5Rn on each floor

(2) Information used in built-in processing

Processing using built-in operations involves matching the following information:

- ON, OFF information for hall calls
  - UP_Xn, DOWN_Xn on each floor
- Dispatching information for each elevator cage
  - Dm, Bm, IN_Xmn in each elevator, and checking competition with the hall call decisions made for the other elevators as will be described in Section 3.2.3, in order to avoid competition.

3.2. Group control system using reinforcement learning

There are, generally, the following dispatching methods: immediate call dispatching method (which involves dispatching cages at the moment when a hall call is generated) and nonimmediate allocation method (which involves periodic dispatching of cages after a certain time period has passed). In the latter method, a decision about dispatching of multiple hall calls generated within a specific time period is made as a result of total estimation of position relationships for multiple cages which can be possibly available. Therefore, this method is advantageous for reducing wait time. In this call dispatching method, several parameters such as

- Elevator status: cage location, movement direction, cage load, cage calls
- Hall call status: hall calls, number of waiting passengers

are substituted for the evaluation function, and on computing anticipated wait time for all possible cages, the best of all options is allocated.

Decision timing for cage dispatching in the proposed method is similar to that in the nonimmediate call dispatching method, with the difference being that as shown in Section 3.2.1 the dispatching decisions are made each time a waiting elevator appears, rather than with specific time intervals as in the nonimmediate call dispatching method.

In addition, the proposed method features capability of flexible adaptation when the distribution patents are captured by reinforcement learning even if it is impossible to know in advance the traffic pattern, main floor locations where a traffic jam might occur, or other parameters. In spite of using a simple technique, this method promises good performance superior to that of the conventional methods, and enables flexible adaptation to changing physical environment and different traffic patterns.

3.2.1 Learning rules

Figure 1 illustrates a state transition diagram for an elevator system, and Table 1 lists contents of each operation causing a state transition. Among the operations listed in Table 1, (1) and (3) correspond to transitions from the waiting state to the next state, that is, to the cage dispatching. This cage dispatching is performed according to the learning rules by a decision on which hall will be responded to by the waiting elevator. As for the remaining operations (2), (4) to (7), they are processed according to the built-in rules.

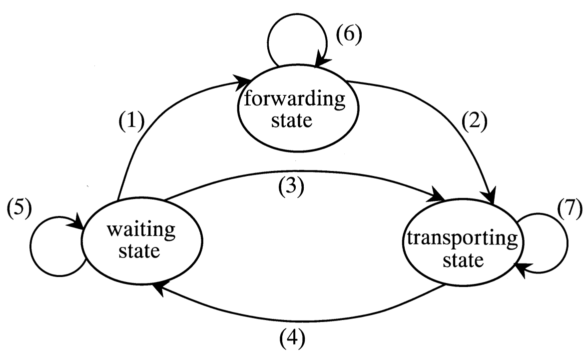

Fig. 1. State transition diagram for the elevator system.

Table 1. Operations causing state transitions in the elevator system

(1)	response to a hall call button (cage allocation)
(2)	boarding completion, response to a cage call button
(3)	response to a hall call button on the same floor (cage allocation), boarding completion, response to a cage call button
(4)	all passengers have gotten off
(5)	holding state
(6)	holding state
(7)	holding state

It is possible that each elevator can make independent action decisions if these learning rules are individually assigned to each elevator.

To illustrate the rules structure, consider three types of information for the Δ hall call button on the first floor:

- Information about ON, OFF status of the hall call button
- Time [seconds] elapsed after the hall call button was pressed
- Number of passengers [people] during the last 5 min

How they are used in the set of rules 1 through 7 is shown in Table 2. Among these rules, No. 7 is intended for forced-dispatching, and is applied in the case when the last number of passengers is high even if the hall call button is OFF.

The rules for the time [seconds] elapsed after pressing the button were set separately for the intervals 10 s and 20 s, since it can be predicted empirically that the waiting time intervals can be subdivided along these boundaries. In addition, the rules for the number of passengers [people] during the last 5 min were set separately for number of people differing by 50 based on the morning rush-hour data collected for 1200 people/hour: the following number of passengers appeared during 5 min in each hall: 90 people in the main floor hall (72 people pressed the △ hall call button on the main floor, and 18 people pressed the ▽ button), and in addition 1 person appeared at each floor, so that with the boundary at 50 people it is possible to detect the hall button locations in which forced distribution is required during the rush hour. For the number of passengers 2400 people/hour, this concept can be similarly applied by doubling the above number of people who appeared during this time interval.

Furthermore, rules 8 through 14 were similarly assigned to hall call buttons at the second floor in the UP direction, which yielded $7[2(N - 1)]$ rules for the total

Table 2. Rules for n-th floor hall call buttons

Rule number	Hall call	Time elapsed after button pressed [sec]	Number of passengers [people] during last 5 min
1	UP_X_1=ON	$0 \leqq$ UP_$T_1 < 10$	UP_5$R_1 < 50$
2	UP_X_1=ON	$0 \leqq$ UP_$T_1 < 10$	UP_5$R_1 \geqq 50$
3	UP_X_1=ON	$10 \leqq$ UP_$T_1 < 20$	UP_5$R_1 < 50$
4	UP_X_1=ON	$10 \leqq$ UP_$T_1 < 20$	UP_5$R_1 \geqq 50$
5	UP_X_1=ON	UP_$T_1 \geqq 20$	UP_5$R_1 < 50$
6	UP_X_1=ON	UP_$T_1 \geqq 20$	UP_5$R_1 \geqq 50$
7	UP_X_1=OFF	UP_T_1=0	UP_5$R_1 \geqq 50$
8	UP_X_2=ON	$0 \leqq$ UP_$T_2 < 10$	UP_5$R_2 < 50$
9	UP_X_2=ON	$0 \leqq$ UP_$T_2 < 10$	UP_5$R_2 \geqq 50$
:	:	:	:
$7(N-1)$	UP_X_{N-1}=OFF	UP_T_{N-1}=0	UP_5$R_{N-1} \geqq 50$
$7(N-1)+1$	DOWN_X_2=ON	$0 \leqq$ DOWN_$T_2 < 10$	DOWN_5$R_2 < 50$
:	:	:	:
$7(2(N-1))$	DOWN_X_N=OFF	DOWN_T_N=0	DOWN_5$R_N \geqq 50$

number of hall call buttons 2(N – 1) present in an N-story building.

These rules had individual weight values, with the initial value set to 100.0 for all rules.

3.2.2. Rule selection

Rule selection is illustrated briefly in Fig. 2. Solid symbols ▲(▼) denote the hall call buttons △(▽) which were pressed. The rule selection procedure is illustrated in Table 3.

3.2.3. Competition check

Through communication with the other elevators, waiting elevators follow a procedure for checking and avoiding competition for the hall call buttons assigned as dispatch candidates according to Table 3. A diagram of communication between elevators is illustrated in Fig. 3. The communication-based competition check procedure is briefly as follows: the elevators ready to be dispatched to respond to a hall call button check through communication with the other elevators whether competition exists regarding that hall call button, and then follow a respective competition avoidance procedure corresponding to the state (waiting, forwarding, transporting) of the other elevators as shown in Table 4, in order to make a decision whether they should respond to the candidate hall call button or delegate the task to another elevator. At this time, elevators are assigned an equipment number.

If the task is delegated to another elevator, then according to the procedure of dispatching decision method shown in Fig. 4 once again a hall call candidate is selected from the applicable rules, and again the competition check of Table 4 is repeated until, finally, a decision is made upon the hall call button for dispatching.

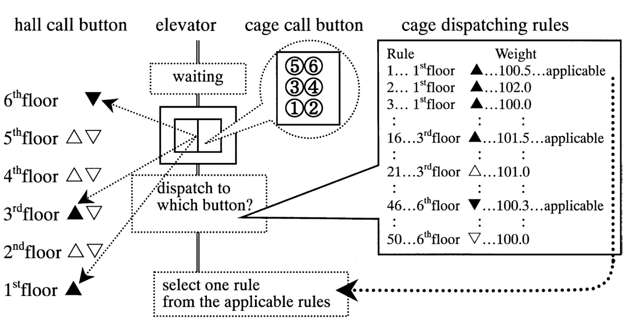

Fig. 2. Concept of rule selection.

Table 3. Rule selection procedure

1) Check applicability of all rules

2) Applicable rules are rearranged in rank by their weight values (if the weights are equal, then the ranks are defined by random numbers)

3) The hall call button to which the 1st rank rule belongs is designated a dispatch candidate

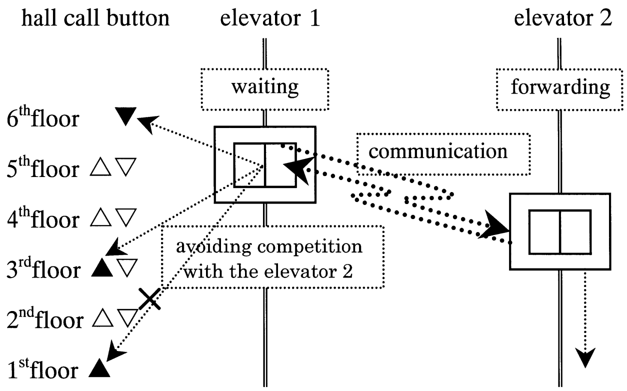

Fig. 3. Diagram of communication between elevators.

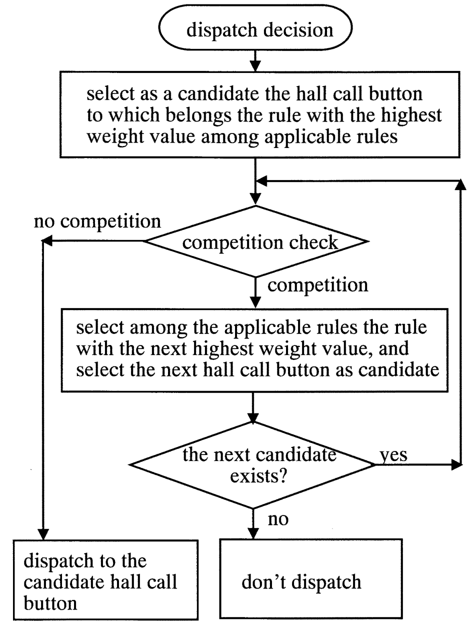

Fig. 4. Elevator dispatch decision procedure.

3.2.4. Learning method

The elevator operations starting from a waiting state to the cage dispatching to completion of passenger transportation to transition again into a waiting state will be designated as one run process.

Let us denote the total waiting time from the start of experiment for the entire elevator group averaged over the total number of passengers as cumulative mean waiting time T [s], and the total number of transported passengers averaged over the total number of runs as cumulative mean number of transported people P [people].

The learning goal consists in converging parameters T and P to specific values.

Each time an elevator completes one run process the rule weight reached as a result of this run (which was used

Table 4. Rules for avoiding elevator competition

```
if (another elevator is in a waiting state)
  if (competition exists with another elevator for a hall call button assigned
      as a dispatch candidate)
    if (its distance to this hall call button is shorter than that of the other
        elevator)
        /* this elevator responds to the hall call */
    else if ((distance to this hall call button is the same as that from the other
             elevator) and (this elevator's equipment number is lower))
        /* this elevator responds to the hall call */
    else
        /* the task is delegated to the other elevator */
  else
        /* this elevator responds to the hall call */
if (the other elevator is in a forwarding state)
  if (the other elevator is being forwarded to this hall call)
        /* the task is delegated to the other elevator */
  else
        /* this elevator responds to the hall call */
if (the other elevator is in a transporting state)
  if (the other elevator is about to stop for this hall call)
        /* the task is delegated to the other elevator */
  else
        /* this elevator responds to the hall call */
```

for the cage dispatching) is updated based on the following evaluation formula (1).

Assuming that the target value of the cumulative mean waiting time T is α [s], the target value of the cumulative mean number of transported people P is β [people], and the run process completed after an elevator m used a rule number r is denoted by s, the updated rule weight at this time can be expressed by $W\_mr(s)$, and the evaluation formula is given by

$$W\_mr(s) = W\_mr(s-1) + \frac{\alpha - T}{k} + \frac{P - \beta}{k} \qquad (1)$$

where k is a constant, which was set as $k = 100$ in the tests.

In Eq. (1), the target values α, β have the following meaning.

- The weight is changed so that the cumulative mean waiting time T becomes smaller than the target value α.
- The weight is changed so that the cumulative mean number of transported people becomes greater than the target value β.

This evaluation formula is used in learning.

Said target values α, β are supposed to be set empirically during the tests for a specific elevator system comprising M elevators in an N-story building. Details of how to set these values are described in Section 4.2.

4. Experiments

This section describes tuning method for the learning parameters in order to achieve reinforcement learning, and shows experimental results obtained from comparison with conventional methods (Hikita and Komaya [5]).

4.1. System specifications

4.1.1. Specifications of cages and building

For comparison with the conventional system [5], the specification conditions in the proposed system were set to the same values. The system was set up in an imaginary standard office building having six functioning cages. In addition, comparison tests were conducted for two cases using two types of passenger data under the assumption of traffic flow corresponding to peak hours.

Specifications of this test system are as follows:

- Number of cages: 6 (carrying 24 people)
- Nominal cage speed: 150 m/min
- Building: 11 stories, main floor 3 levels
- Height per floor: 3.8 m
- Maximum acceleration: 1 m/s^2

The elevator speed controls were designed for outputting sufficient speed within a range suitable to satisfy the foundation and safety conditions [6], with the equivalent acceleration time t_1 and the equivalent speed time t_2 tuned in accordance with the running distance and with the speed-acceleration diagram shown in Fig. 5.

In addition, the door open–close time and the passengers getting on–off time were set as follows:

- Door width: 900 mm (double-sided double sliding doors)
- Door open–close time: 2 s each
- Getting on–off time: 0.8 s per person, 0.95 s per person

4.1.2. Passenger data in conventional system

This section describes passenger data used for tests in the conventional system and in the proposed system.

1. Preliminary tests for examining general efficiency during morning rush-hour operation

For better simulation reliability, two types of passenger data were used: 1200 and 2400 people/h (in both cases 90% were riding from the main floor and 10% from the other floors) assuming that the flow corresponding to peak hours lasted for 3 h although in common buildings the rush-hour peak time lasts only 15 to 30 min. The passenger data were generated using random numbers, and their probability distribution was based on a specific passenger OD distribution. The passenger OD distribution reflects the moving rates (or its proportion numbers) of the passengers between floors (or zones) in the building.

2. Tests for examining system efficiency

To test the system efficiency we obtained five groups of 1-h-duration passenger data based on actual data acquired by measurements for 5 days (Monday to Friday).

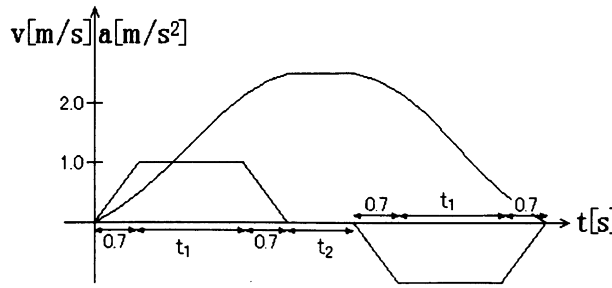

Fig. 5. Speed–acceleration diagram.

4.1.3. Passenger data in proposed system

Details of the tests and passenger data for the proposed system are explained below.

1. Tests for learning traffic flow pattern

To achieve the goal of tuning the learning parameters in application to the traffic flow patterns and learning the rule weights, 3-h learning sessions were conducted using the two types of passenger data—1200 and 2400 people/h—that were used in the conventional system tests (1).

2. Tests for examining system efficiency

For correspondence with the conventional system (2), to imitate actual run operation the same passenger data and passenger numbers 1200 people/h (2400 people/h) were used as in the proposed system (1), as well as the same passenger OD distribution: five sets of 1-h-duration passenger data were used with only the random number (random number generation method) modified, and conducted 1-h-long tests for each case.

For comparison of the results for system (2) with those for the conventional system, simulation tests were conducted with the same data fed also into the conventional system.

4.1.4. Conventional system rules

An example of rules used for forced-dispatching in simulation tests for conventional system is shown in the study by Hikita and Komaya [5]. This study shows an actual example of making decisions upon the number of forced-dispatched elevators. However, thinking that other efficiently working rules besides those shown must exist, the authors conducted tests aiming at decision upon selecting elevators for forced-dispatching, with additional rules defining how with several elevators in waiting state dispatching is made from the nearest location on the main floor upon considering their location relationships.

4.2. Adjustment of learning parameters

Traffic pattern learning tests were conducted according to Section 4.1.3 (1): they involved a 3-h-long learning session with given passenger data 1200 people/h and with the target values α, β in Eq. (1) set to the following numbers as in (1) to (9). The tests were conducted for different combinations of α and β with β set to 3 and α to the following numbers:

(1) $\alpha = 6$, (2) $\alpha = 7$, (3) $\alpha = 8$,
(4) $\alpha = 9$, (5) $\alpha = 10$, (6) $\alpha = 11$,
(7) $\alpha = 12$, (8) $\alpha = 13$, (9) $\alpha = 14$

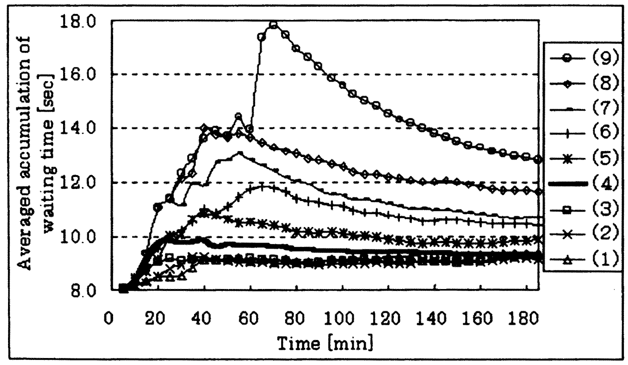

Fig. 6. Cumulative mean waiting time (1200 people/h) during learning.

Figure 6 illustrates corresponding variations of the cumulative mean waiting time *T*.

As shown in Fig. 6, the learning curves generated for the *T* versus time transition, converged for all of settings (4) to (6). [As for settings (8) and (9), the learning was continued for another 180 min, with convergence reached at 11.5 and 12.5 s, respectively.] It was discovered that the learning curves featured larger variation in the direction from (4) to (9), and the convergence value became higher, as well. For settings (1) to (3) learning could not be confirmed.

Based on these results it was decided upon using the setting α = 9, β = 3 for the passenger data (traffic pattern) 1200 people/h.

Figure 7 shows transition of waiting time averaged over 5-min intervals for this setting (average waiting time measured, typically, for each 5-min time interval).

The number of learning runs from the test start moment to the end moment 3 h later for the entire elevator group was 1492, and *T* converged at 9.34 s, with the cumulative average number of transported people *P* converged at 2.67 people.

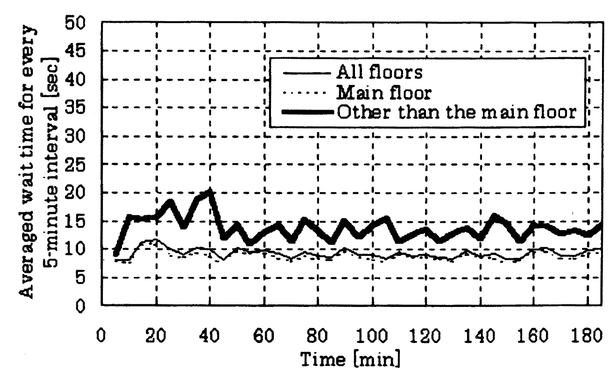

Fig. 7. Waiting time averaged for each 5-min interval during learning (1200 people/h).

Figure 7 reveals that learning can be completed in about 40 min from the learning start moment, after which a steady operation is continued.

Next, a similar tuning of learning parameters was conducted for the passenger data 2400 people/h, for which the settings $\alpha = 15$, $\beta = 12$ were used. The number of learning runs from the test start moment until its completion 3 h after for this setting was 1193, and T converged at 14.42 s, with the cumulative average number of transported passengers converged at 8.78.

4.3. Test conditions

Test conditions used for examining the efficiency of the system in Section 4.1.3 (2) are as follows:

1. We used five sets of 1-hour-duration passenger data: namely, the 1200 people/h passenger data used in Section 4.2, with the total number of passengers and the passenger OD distribution the same, and only random numbers changed.
2. The initial learning rules weight was set to the rule weight captured from the 3-h-long learning with the passenger data 1200 people/h as in Section 4.2.

The tests were conducted with the learning function set to ON.

In addition, the above-described five sets of passenger data were fed into the conventional system, for which tests were conducted with a similar imitation of environment variations.

Next, similar tests were conducted for the passenger data 2400 people/h.

4.4. Test results

The test results for the conventional system and the proposed system were compared in the case of given passenger data 1200 people/h. In each test waiting time values averaged over 5-min intervals were obtained for all floors, main floor, and other than the main floor, and these waiting times were averaged over 1-h intervals from the test start until the end. In addition, these values were averaged over the five sets of test data. The results are shown in Fig. 8.

In comparison with the conventional system, the average waiting time was observed to improve by 20.0% for all floors, by 14.9% for the main floor, and by 40.4% for the other floors.

Considering that during rush hour 90% of passengers appear on the main floor, excessive improvements directed to the other floors would result in insufficient main floor coverage, with overall deterioration of the waiting time results; in spite of this complication, the proposed method not only allowed improvement of waiting time on the main floor, but also significantly reduced waiting time on the other floors.

Next, for the case with given passenger data 2400 people/h similar tests were conducted on the basis of rule weight values acquired during a 3-h learning session. Results of comparison with the conventional system are shown in Fig. 9.

In comparison with the conventional system, the average waiting time improved by 16.1% on all floors, by 13.3% on the main floor, and by 30.2% on the other floors.

In the case with given passenger data 2400 people/h, which is considered a congestion situation, similar to the case with 1200 people/h, the proposed method allowed in addition to the waiting time improvement on the main floor a significant reduction of waiting time on the other floors.

4.5. Discussion

Results of tests conducted using the rules acquired in Section 4.2 with the passenger data 1200 people/h (2400 people/h) according to Section 4.3 demonstrated as described in Section 4.4 that in the proposed system in addi-

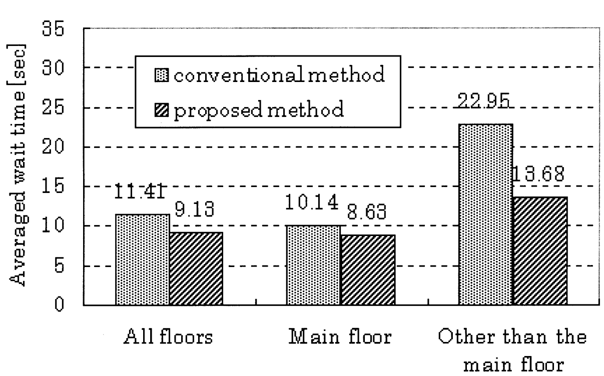

Fig. 8. Comparison between the conventional and proposed systems (1200 people/h).

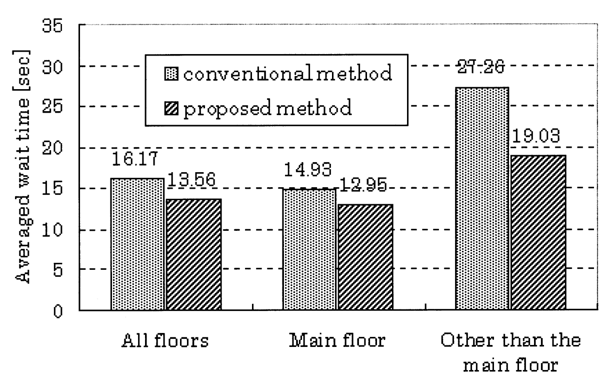

Fig. 9. Comparison between the conventional and proposed systems (2400 people/h).

tion to waiting time improvement on the main floor, waiting time on the other floors was also significantly reduced. The reasons will be discussed taking into account the following issues relating to efficient learning rules acquired as shown in Section 4.2:

- weight values of rules learned by elevators
- rules of elevator response
- polarization of elevators

4.5.1. Weight values of rules learned by elevators

Each elevator is provided with learning rules for responding to a call hall button as shown in Table 2. Let us examine learning content of these rules.

Figure 10 shows the learned rule weights for the hall call Δ button on the main floor after learning.

This graph shows the tendency to selection of the same rules by all elevators; however, they have different rule weights which reflect the learning depth.

First, rule 18 for the hall call △ button on the main floor has the largest weight value for all elevators, and it is evident that it provides good results if response is achieved within 10 to 20 s after pressing the hall call △ button.

It can be explained that in the case of 1200 people/h a passenger appears every 3 s, with 90% of the total number of passengers on the main floor, and 80% of them pressing the hall call button △. In other words, a passenger appears every 4.17 s at the hall call button △, and, as shown in Fig. 11, when 10 to 20 s elapsed after the first one arrived in the main floor hall and pressed the hall call button Δ the total number of people waiting in queue reaches 3 to 6, with the average waiting time for each passenger about 6 to 11 s. As the elevator arrives, the door opens, and the next moment after the elevator starts moving an ideal relationship between the number of transported passengers and the average waiting time is obtained. Therefore, it is evident that this rule's weight increases.

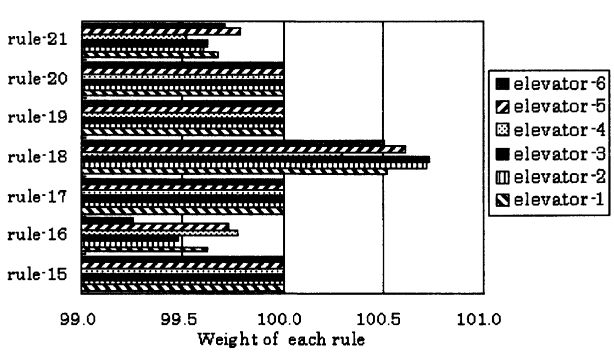

Fig. 10. Rule weights for the hall call button △ on the main floor (1200 people/h).

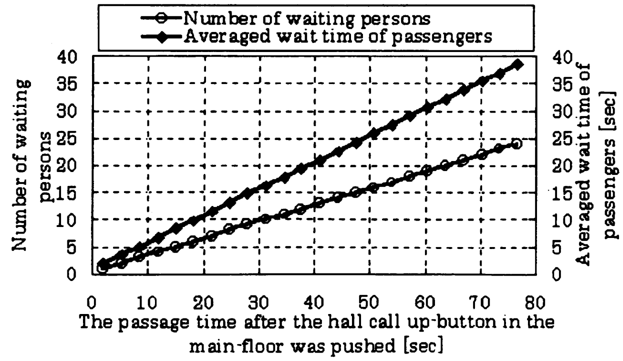

Fig. 11. Variation of averaged waiting time versus time elapsed upon pressing the hall call button △ on the main floor.

As for the other rules, in the case of rule 16 dealing with the time not exceeding 10 s the weight value fell from the initial value. Then, the forced-dispatch according to rule 21 also fell from its initial value.

Regarding floors other than the main floor, for both △ and ▽ buttons the first rule is selected as superior. However, considering that as a result of selecting these rules and dispatching the waiting time exceeds the target value, their weight values are reduced from the initial values.

4.5.2. Elevator response rules

Let us examine in detail response rules for individual elevators.

Assuming that the hall call buttons responded by an elevator can be subdivided into the following types:

- Hall call button ▽ on a floor above the main floor
- Hall call button ▽ on the main floor
- Hall call button ▽ on a floor below the main floor
- Hall call button △ on a floor above the main floor
- Hall call button △ on the main floor
- Hall call button △ on a floor below the main floor

Let us examine simply the number of hits regardless of the weight values. Figure 12 shows types of hall call buttons responded by each elevator, and Fig. 13 shows the number of people transported in each elevator.

It can be seen from these graphs that the number of passengers transported by each elevator was about 600 for all of them, and the following trends were present for different types of the hall call buttons responded by each elevator.

4.5.3. Polarization of elevators

By examining the trends for hall call buttons responded by each elevator one can see that six elevators are

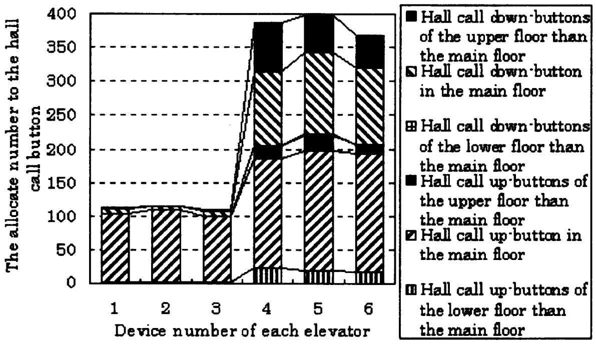

Fig. 12. Breakdown of hall call buttons responded by each elevator (1200 people/h).

subdivided into the two types by their learning characteristics (see Fig. 12).

First, in one group comprising elevators 1, 2, and 3, all three elevators exhibited the same trend with about 90% responding to the hall call buttons △ on the main floor, and about 5% responding to the hall call buttons ▽ on the main floor. The remaining 5% responded to the other hall call buttons. The cumulative average number of transported passengers P was about 5.5.

Next, all three elevators in the group comprising elevators 4, 5, and 6 exhibited the same trend with about 45% responding to the hall call buttons △ on the main floor, about 30% responding to the hall call buttons ▽ on the main floor, about 15% responding to the hall call buttons ▽ on the other floors, and about 10% responding to the hall call buttons △ on the other floors. The cumulative average number of transported passengers P was about 1.6.

Thus, the former group learned properties suitable for focused response on the main floor, and for providing transportation to a crowd of people by a small number of runs, while the latter group learned properties suitable for widely distributed response to the floors other than the main floor, for accurate response to even a small number of people, and for providing multiple transportation runs. Such polarization can be explained as follows: during competition the system preferences were set to the group characterized by lower equipment numbers, so that when competition occurred for response on the main floor with a high event rate, the group characterized by lower equipment numbers had response priority. This resulted in different hall calls suited specific elevators, and created specification, which is a feature of multiagent systems. Therefore, it seems to be the reason why the proposed system proved to be more efficient than the conventional system.

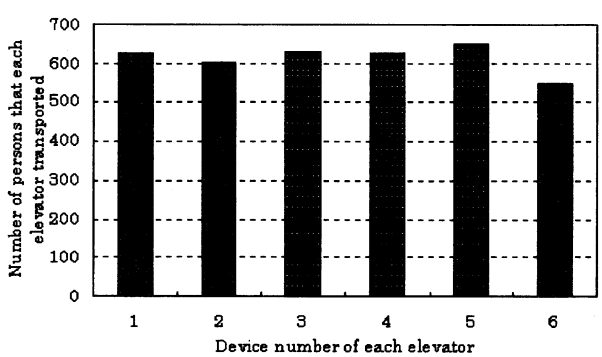

Fig. 13. Number of passengers transported by each elevator (1200 passengers/h).

5. Comparison of Cases with Changed Environment

Let us assume that during rush-hour congestion one elevator of the six available stopped due to malfunctioning, which caused an environment change requiring response from the remaining five elevators. For higher test reliability we used the passenger data under the assumption that the rush-hour peak time lasted 3 h. Within these 3 h, the first hour of traffic was covered by the usual six elevators, and the remaining 2 h by five elevators.

5.1. Settings for changed environment

We will demonstrate that in the case of changed environment the proposed method provides flexible response by learning. The test conditions for changed environment are as follows:

1. As passenger data, five sets of 3-h-duration passenger data were used with the total number of passengers and OD distribution of passengers the same as in the case of 1200 people/h according to Section 4.2, and modified only the random numbers.
2. The initial state of the test system was set according to the specifications shown in Section 4.1.1, and 1-h runs were conducted in the state with six functioning elevators.
3. After 1 h from the test start one of the six elevators was stopped due to assumed malfunctioning, and 2-h-long tests were conducted with the remaining five elevators.

For initial values of the learned rule weights we used the rule weights resulting from a 3-h test with the passenger data 1200 people/h, and with the learning function left ON.

In addition, tests with similar imaginary change of environment were conducted for the conventional system with the given five sets of passenger data described above.

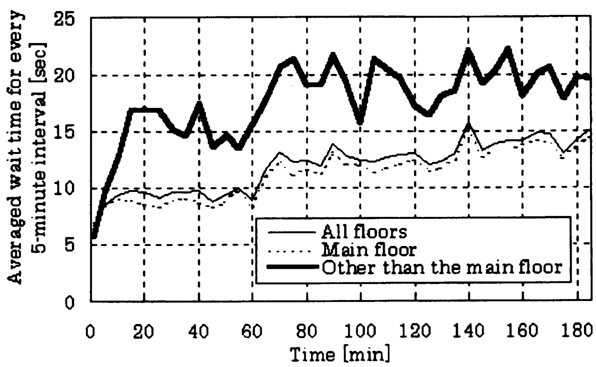

Fig. 14. Waiting time averaged over 5-min intervals obtained by the conventional method (with change of environment, 1200 people/h).

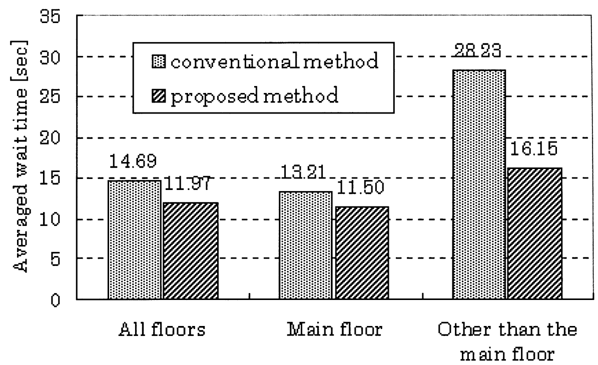

Fig. 16. Comparison between conventional and proposed methods (with environment change, 1200 people/h).

5.2. Test results

The results for waiting time transition averaged over 5-min intervals for all floors, the main floor, and the other floors were averaged for the five sets of test data, and displayed on broken-line graphs. The results for the conventional system are shown in Fig. 14, and those for the proposed system, in Fig. 15.

It can be seen from a comparison of these results that the proposed system of Fig. 15 provided flexible response and steady operation even when an environment change occurred 1 h after the start of the tests.

In addition, for the 5-min averaged waiting time values obtained for all floors, the main floor, and the other floors, we averaged each over the 1-h time interval from the moment 1 h after the change in environment until the moment 2 h after, and then the obtained values were averaged over the five sets of the test data and displayed on bar graphs. Figure 16 compares results for the conventional system and proposed system.

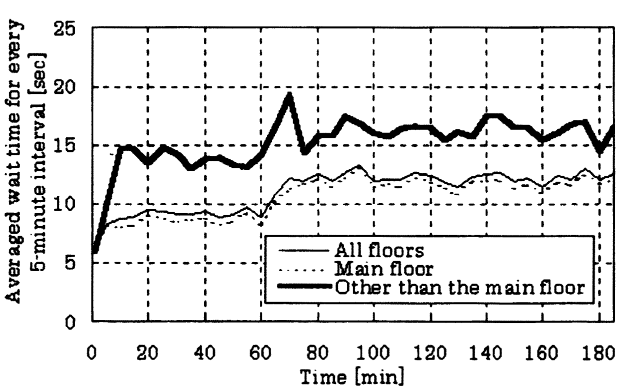

Fig. 15. Waiting time averaged over 5-min intervals obtained by the proposed method (with change of environment, 1200 people/h).

The graphs show that compared with the conventional system the average waiting time improved by 18.6% for all floors, by 12.9% for the main floor, and by 42.8% for the other floors. Even in the case of changed environment, the proposed system exhibited a significant reduction of waiting time on the floors other than the main floor, and the waiting time on the main floor was also reduced.

5.3. Discussion

Let us discuss why good results were obtained using the proposed system. In the case without environment change (all six elevators functioning), the average number of transported passengers during 1 h was 2.67 people/h, and the rule selection number was 83.8 per hour. In the case of changed environment, the average number of transported passengers was 3.61 people/h, and the rule selection number was 74.4 per hour. It is evident that in accordance with the environment change (reduced number of elevators) the rule selection number was reduced, and the number of transported passengers per hour increased, the system was able to provide flexible response. Regarding the trends of efficiently working rules, they were the same in both cases: one can see that the learned rules acquired in Section 4.2 could be applied without change.

6. Conclusion

In the case without environment change, the proposed method in comparison with the conventional method provided for a significant service improvement on the floors other than the main floor, which is considered the most difficult task during morning rush-hour traffic, without sacrificing the main floor service. The overall service was also improved.

In addition, in the case with environment change, the proposed system provided for flexible response, with the same good results as for the case without environment change.

This study examined the efficiency of the proposed method during morning rush-hour traffic; however, since it is based on a flexible learning method, it can be expected to provide flexible response also to different traffic patterns such as lunchtime or end-of-day traffic.

Acknowledgments. The authors express their gratitude to Mr. Misawa of the Graduate School of Natural Science and Technology at Kanazawa University, and to research associate Dr. Nanbo of that department for their guidance in this study.

REFERENCES

1. Hirasawa K, Kuzunuki S, Iwasaka T, Kaneko G, Kawatake K. Hall call assignment in elevator supervisory control. J Electron Inf Commun Eng Japan 1979;99:1–6.
2. Kurosawa K, Hirasawa K, Kuzunuki S, Yoneda K, Sakai Y, Haginaka H. Intelligent and supervisory control for elevator group (Part I: Hall call assignment logic). J Inf Process Soc Japan 1985;26:278–287.
3. Kurosawa K, Miura M, Hirasawa K, Yoneda K, Sakai Y, Haginaka H. Intelligent and supervisory control for elevator group (Part II: Learning logic). J Inf Process Japan 1987;28:299–309.
4. Hikita S, Komaya K. A new elevator group-supervisory control system using fuzzy rule-base. J Soc Instrum Control Eng 1989;25:99–104.
5. Hikita S, Komaya K. Elevator group control during morning rush-hours. J Soc Instrum Control Eng 1989;31:1896–1904.
6. Hori T, Hayashi K. Design of elevators and escalators for construction and equipment engineers. Gijutsu Shoin; 1994.

AUTHORS (from left to right)

Yasuhiro Ogoshi (member) received his B.E. degree from Tamagawa University in 1991 and joined Niigata Engineering Co., Ltd. In 1996, he completed the first part of his doctoral course in information science at the Japan Advanced Institute of Science and Technology and joined I-O Data Device, Inc. In 1997, he became a research assistant at Fukushima Women's College. In 1998, he started the last course of his doctoral studies at Kanazawa University, and became a lecturer at the same college in 1999. In 2001, he received a Ph.D. degree in engineering, and became a research assistant on the Faculty of Engineering at Toyama University. He is engaged in studies relating to multiagent research. He is a member of the Institute of Electrical Engineers of Japan, the Information Processing Society of Japan, the Japanese Society for Artificial Intelligence, and the Japan Society for Software Science and Technology.

Haruhiko Kimura (member) received his B.E. degree from Tokyo Denki University in 1974, and M.E. and D.Eng. degrees in information engineering from Tohoku University in 1976 and 1979. He then joined Fujitsu, Ltd. In 1980, he became a lecturer at Kanazawa Women's College, and in 1981 an associate professor; an associate professor on the Faculty of Economics at Kanazawa University in 1984, and on the Faculty of Engineering in 1992, and has been a professor since 1994. He has been engaged in studies relating to optimal code transformation, and speed improvements in production systems. He is a member of the Information Processing Society of Japan and the Japanese Society for Artificial Intelligence.

AUTHORS (continued) (from left to right)

Sadaki Hirose (member) received his B.E. degree from Toyama University in 1974, and M.E. and D.Eng. degrees in information engineering from Tohoku University in 1976 and 1980. He then joined Fujitsu Laboratories. In 1984, he became an associate professor on the Faculty of Engineering at Kanagawa University, in 1989 an associate professor on the Faculty of Engineering at Toyama University, and since 1998 has been a professor. He has been engaged in studies relating to automaton and formal language theory, algorithmic analysis, theory of complexity of computing, and similar topics. He is a member of the Information Processing Society of Japan.

Nobuyasu Osato (member) received his B.E. degree from Kyushu Institute of Technology in 1974, M.E. degree in information engineering from Tohoku University in 1976, and Ph.D. degree in engineering. In 1998, he became a professor on the Faculty of Engineering at Nihon University, and since 2002, he has been a professor on the Faculty of Information Science and Technology at Osaka Institute of Technology. He is engaged in R&D studies being conducted, for example, at the Musashino Electrical Communications Laboratory relating to symbol processing systems, intelligent robots, and distributed cooperating systems. In 1990 he received the annual author award from the Information Processing Society of Japan. He is a member of the Information Processing Society of Japan and IEEE.