Proceedings of the 2000 IEEE
International Conference on Robotics & Automation
San Francisco, CA • April 2000

Dynamic Scheduling Approach to Group Control of Elevator Systems with Learning Ability

Yuan-Wei Ho and Li-Chen Fu
Department of Computer Science and Information Engineering
National Taiwan University, Taipei, Taiwan, R.O.C.

Abstract

In this paper, a hybrid model of a multiple elevator system is proposed, consisting of a color-timed transition Petri net (CTTPN) model and a set of control rules implemented via the so-called control places in the CTTPN model. The Petri net model is a highly modulized structure, whose constituent modules can be classified into four types: Call Management Module, Loading/Unloading Module, Basic Movement Module and Direction Reversing Module. The whole complete model is a combination of the copies of the above four modules. Since the firing sequences of the CTTPN equate the evolution of the modeled system, they can be regarded as a schedule. A dynamic scheduling with learning ability is proposed to obtain the desirable schedule. A new concept of control places is also introduced in the proposed model so as to make the modeling more precise and to reduce the reachability graph more efficiently. To show the feasibility of the proposed method, an emulator in Elevator Control Kernel and Elevator Scheduler Kernel are constructed for demonstration.

1. Introduction

An elevator system is used to supervise multiple elevators and to ensure that they are operated efficiently. The elevator group control system must decide which of several elevator cars should respond to a call made by passengers, in order to provide the highest possible level of service. To build a good elevator system for solving the elevator scheduling problem, we need to build an elevator model first to represent the cooperation of multiple elevators. Petri net has evolved into an elegant and powerful graphical modeling tool for asynchronous, concurrent event-driven system and, therefore, is chosen here as a suitable modeling tool.

However, the elevator system is usually involved with human behaviors and preferences. Very often, there may not exist an optimal scheduling solution and, hence, the control system must be flexible enough to handle all possibly incoming situation. In view of this, a hybrid Petri net model is thus introduced in this paper to serve the purpose, i.e., besides the conventional color-timed transition Petri net model, several heuristic rules and control places are introduced to enrich the flexibility of the aggregate model. Upon the occurrence of an external event, the model will search for the feasible rules to adjust the marking of the resulting Petri net, whereby the hybrid model will run properly to produce an adequate scheduling result to handle that situation for service requirement.

Petri net [1, 2] is a graphical and mathematical modeling tool applicable to many systems. They are promising for describing and studying information processing systems that are characterized as being concurrent, asynchronous, distributed, parallel, nondeterministic, and/or stochastic. Petri net is a useful tool for modeling [3, 4], simulation, and flexible manufacturing sytems.

Researchers from many scientific disciplines are designing *artificial neuronal networks* (ANNs) [5, 6] to solve a variety of problems in pattern recognition, prediction, optimization, associate memory, and control [7, 8]. Conventional approaches have been proposed for solving these problems. Although successful application can be found in certain well-constrained environment, none is flexible enough to perform well outside its domain. ANNs provide exciting alternatives, and many applications could benefit from using them.

Due to the fast development of computer technology, integrated circuit (IC) and micro-computer are widely used in modern elevator system. The computer technology introduces the artificial intelligence (AI) technique into the elevator group supervisory control system [9, 10]. The fuzzy control logics [11, 12] and expert systems are used to analyze the traffic pattern and to predict the possibly incoming call requests [10, 13]. Moreover, the statistics approximation is applied to estimate hall call waiting [14]. Conceivably, the utilization of the AI and expert systems in the context of elevator system control is an indispensable trend.

2. Modeling and Scheduling

2.1 Systematic Modeling

Problem Description

0-7803-5886-4/00/$10.00© 2000 IEEE

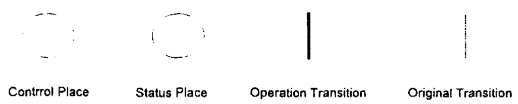

Figure 1: Types of places and transitions

In a typical elevator system, there are two types of calls. The *hall call* is given through buttons on the hall of the building, and the *car call* is given by the passengers inside the elevator. An elevator system has a pair of hall call buttons on each floor, one for *up hall call* and the other for *down hall call*. As soon as a hall call button is pressed, the elevator system must register the hall call, and selects and assigns an elevator to serve the hall call. After serving the hall call, the passenger should press the car call button to register his destination floor and the elevator must move up/down to stop at the destination floor. When all calls are served, the elevator gets look to an idle state. We can park an idle elevator at a specific floor to reach the goal of better performance, and then repeat the above operation again.

In this paper, we select the CTTPN to model our elevator system. In this CTTPN model, all the places and transitions can be classified into two categories listed below as shown in Fig. 1:

1) **Control places**: These places represent the signals received from the world outside the elevator system or from the elevator system.

2) **Status place**: These places are used to record some specific information about the current elevator system.

3) **Ordinary transition**: These transitions are the same as the ordinary Petri Net. The transition will be fired immediately when the transition is enabled.

4) **Operation transition**: The firing of an operation transition proceeds as in an ordinary Petri Net but with a modification: the tokens are added to the output places only after pre-assigned period of times depending on the color of the input place, which however are changeable from a firing to another. Note that the colors of the tokens in the places before and after the operation transitions can be different.

Due to the complexity of describing various kinds of behavior of a group of elevators, it is hard to construct the Petri net model directly. Therefore, we first build four kinds of macro modules so that the final overall Petri net model can be constructed from these macro modules.

Call Management Module

Call Management Module is mainly used to manage the car calls and the hall calls. When the system finds that

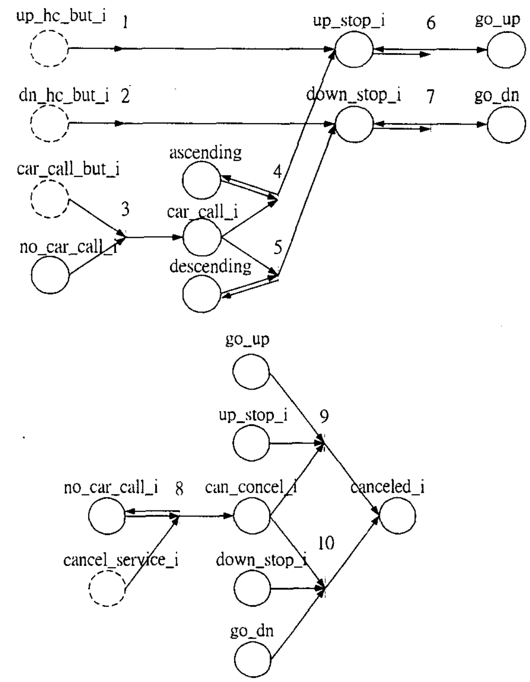

Figure 2: Call Management Module

service of some hall call may be a better choice, it is a good opportunity to cancel the service and re-assign the hall call to another elevator car. We show the Call Management Module in Fig. 2.

Loading/Unloading Module

The Loading/Unloading Module describes the event where passengers get into or get out of the elevator car. An elevator must be parked at the floors designated by the car calls, but it may not stop at the floors designated by the hall calls for some reasons. The Loading/Unloading Module is shown in Fig. 3. Places in different modules but with the same notations represent that they are shared among those modules.

Basic Movement Module

The Basic Movement Module describes the movement activities of each elevator between floors. The Basic Movement Module is shown in Fig. 4. By repeating this strategy for every suffix *i*, we can eventually construct the full movement model of an elevator car. Then, by adding a prefix serial number, we can represent different elevators. For example,"3.floor_7" represents the #3 elevator car is parked at or pass through the 7-th floor. All places without suffix represent that they are shared among all elevators.

Direction Reversing Module

Direction Reversing Module describes the timing to reverse the moving direction of an elevator car, which is

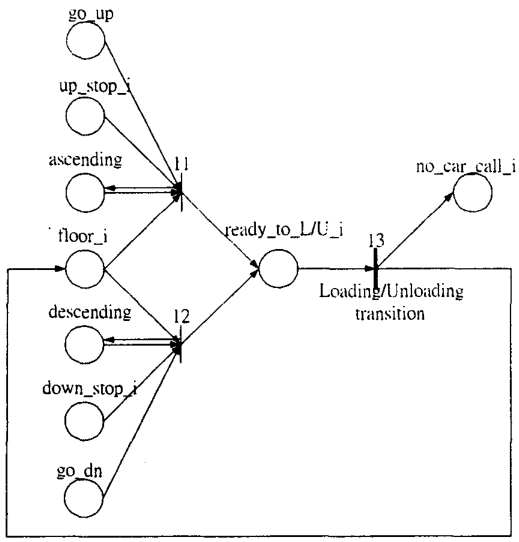

Figure 3: Loading/Unloading Module

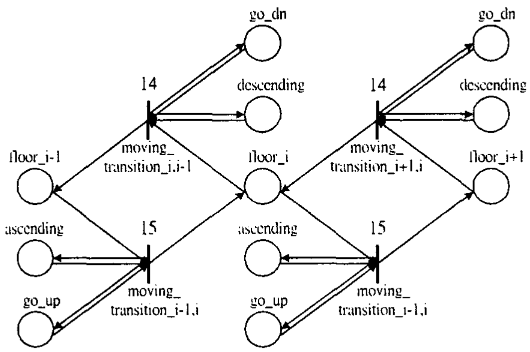

Figure 4: Basic Movement Module

shown in Fig 5. To check whether an elevator car can reverse its moving direction, we must check all the hall calls and car calls in its moving direction. In order to simplify the model, we use simply the unbounded places "go_up" and "go_dn" to be the checking model.

A Simple Example : Modeling with Modules

In Fig. 6, we show a simple example. There are three elevators in a building with four floors in the example. The rectangles denoting "BM" represents the Basic Movement Module, the rectangles denoting "DR" represents the Direction Reversing Module, and the rectangles denoting "L/U" represents the Loading/Unloading Module. But the Call Management Module consists of two kinds of rectangle, respectively, denoting "car call" and "hall call". The circle, "floor", represents a **floor_*i*** place for some *i*. We use the place **floor_*i*** to connect the Basic Movement Module, the Loading/Unloading Module, the Direction Reversing Module, and the Call Management Module. Similarly, we use places "hall call" to integrate different elevators into a group.

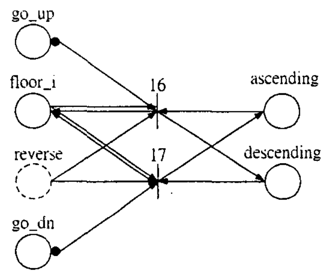

Figure 5: Direction Reverse Module

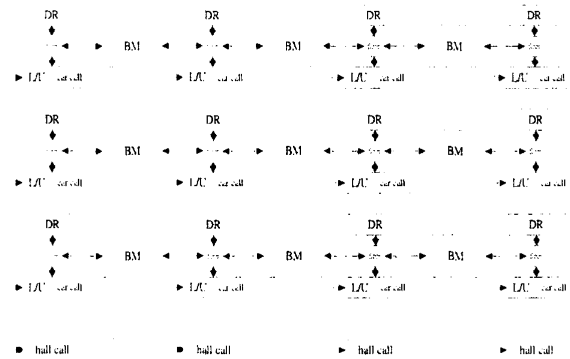

Figure 6: A simple example of three elevators in a building with four floors

2.2 Rules of the Model

To combine the rules and Petri net model, we use the control places as an interface between the rules and the Petri net model.

Control Rules and Control Places

Rule 1) Assign tokens to *Reverse*:
When there is no more car call or hall call assignment along the current moving direction of an elevator, the system can assign a token to the place *reverse* to reverse its moving direction.

Rule 2) Assign/remove tokens to or from *Button* :
Button places include *up_hall_button_i*, *down_hall_button_i*, and *car_button_i places*. When a hall/car button is pressed and that call has not been registered, the system will assign a token to the corresponding place.

Rule 3) Assign/remove tokens to or from *Canceled_event* :
When the system finds an assignment better than the one predicted previously, the system will assign a token to the corresponding place cancel_service_*i* to cancel the previous assignment.

Another advantage of taking control places is their "flexibility". To adapt to various traffic conditions and different performance criteria, we can elaborately assign the necessary tokens to the control places and guide the evolution of the Petri net model to suit our need.

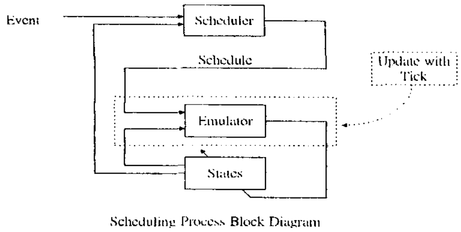

Figure 7: Block diagram of the dynamic scheduling.

2.3 Dynamic Scheduling

In this paper, we want to generate the real-time schedule of an elevator system. The scheduler try to minimize the passengers' average waiting time, the long wait probability, the riding time, the long riding probability, and power consumption.

We propose an event-driven elevator system to handle every event of an elevator system. Our events include *hall calls*, *car calls*, *end-of-loading* and *tick*. "*tick*" is an event that is triggered every fixed time interval, say, a second, by the system clock. Here, we also use *tick* as the time unit for tracing the transition of firings of our CTTPN model. The block diagram of the system is shown in Fig. 7.

To provide satisfactory performance, we embed the sense of satisfaction of passengers into our elevator system, estimate detailedly the cost by simulation over our Petri net model, and refine the dispatching by an refinement algorithm.

Cost Functions

We try to simulate the real circumstance of elevator system by estimating cost with some accumulative cost functions. We take waiting time, riding time, and power consumption as parameters of our cost estimation. The cost of waiting is piecewise linearly increasing with time, and the cost of riding is considered with both the riding time and the moving distance. We define a aggregate weight cost:

cost = waiting cost + riding cost + power cost

Hall Call Assignment

In our system, when a hall call occurs, we first assign the hall call to an elevator randomly. Then, we simulate the consequence of the assignment by evolving from the current marking to a goal marking over our Petri net model, and record the total evolution cost. After that, we assign the hall call to another elevator and repeat the above steps. After the exhausted simulations, we choose the elevator with the minimum cost as our preferable dispatching. Fig. 8 shows the flow of the hall call assignment.

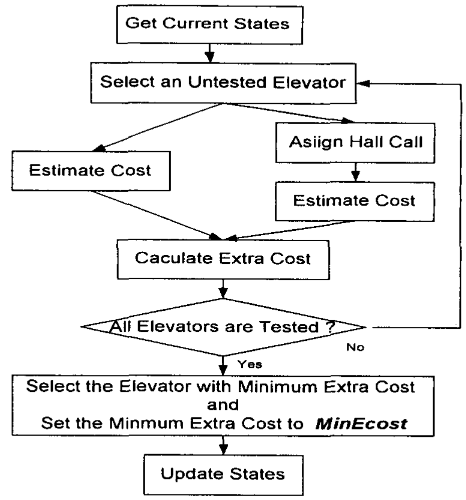

Figure 8: Flow of the hall call assignment.

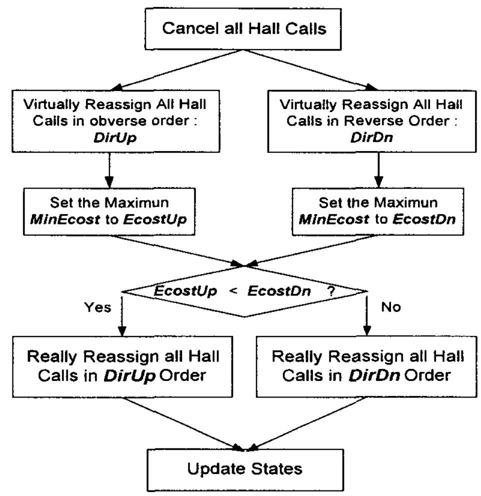

Figure 9: Flow of the refinement algorithm.

Two-Way Refinement

When any of the aforementioned events occurs, the prediction made in hall call assignment should be updated, and the dispatch should be updated as well. To resolve the above problem, we first make an implicit assumption that, whenever an event occurs, the influence to the optimality of the previous schedule should be local. Thus, we try to correct our schedule to approach the optimal schedule. However, such assumption may not adequate, but it is hardly possible to schedule the whole system again in a real-time system. Hence, we propose an algorithm that focuses on the solution set near our previous schedule to search for a refined one. Fig. 9 shows the flow of the *Two-Way Refinement Algorithm*. The algorithm is efficient and economic, because it only takes constant memory and occupies little CPU times. Such desirable performance can be observed in the simulation results.

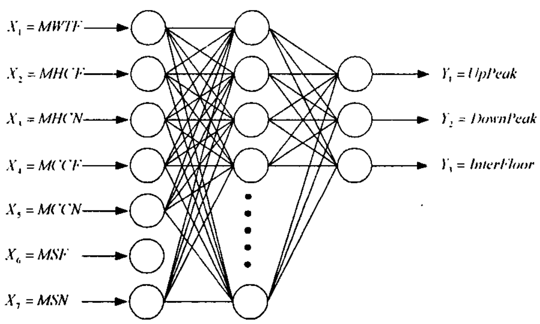

Fig. 10: Network structure EBP

3. Learning Ability

Why Back-Propagation (BP)?

Back-Propagation(BP) is the most popular neural network. It has been applied successfully in a variety of diverse areas. *Prediction* and *classification* are the main tasks of BP. In the elevator system, there are various types of passenger traffic, thus we want to *classify* traffic pattern and then *predict* the floor that an idle elevator should be parked at.

Back-Propagation (BP) applied in Elevator System

The most important criterion of the elevators system is *waiting time*. Besides speeding up the elevator, learning is a better choice to minimize the waiting time. If the scheduler could know the traffic pattern by analyzing the current statistic data, it would park the elevator at the proper floor to minimize the next passenger's waiting time. This is why elevators system should learning. We construct a network structure EBP (7 × 10 × 3) as Fig. 10 .

- There are seven input variables in input layer :

1. **MWTF**: The floor with max. waiting time.
2. **MHCF**: The floor with max. hall call.
3. **MHCN**: The number of hall calls at the floor **MHCF**.
4. **MCCF**: The floor with max. car call.
5. **MCCN**: The number of car calls at the floor MCCF.
6. **MSF**: The floor with max stops.
7. **MSN**: The number of stops at the floor MSF.

- There are three output variables in the output layer :

1. **Up Peak Pattern**: If $Y_1 = u$, the traffic pattern is Up Peak from floor 'u'.
2. **Down Peak Pattern**: If $Y_2 = d$, the traffic pattern is Down Peak to floor 'd'.
3. **Inter Floor Pattern**: If $Y_3 = 1$, the traffic pattern is Inter Floor.

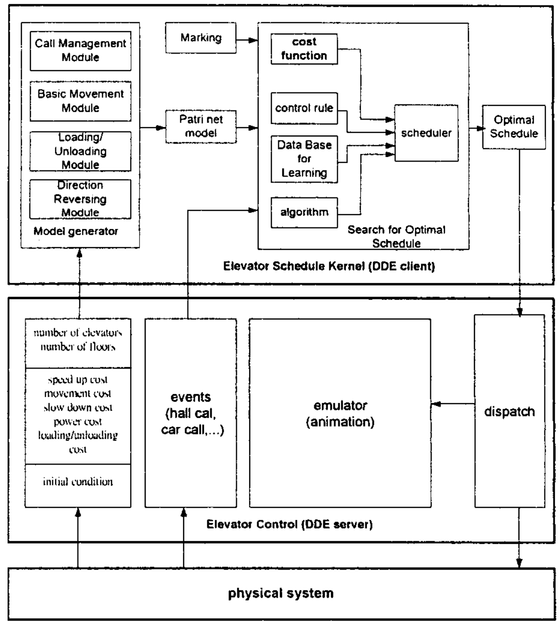

Figure 12: The hierarchical view of the elevator system

For example, if the output vector of the neural network is [1,0,0], we can identify the running traffic pattern is Up Peak from floor '1'. Of course, the scheduler should park the idle elevator at floor '1'..

Mode 1: Scheduler catches the output data of neural network from 20 minutes ago. If the traffic pattern system is running is wrong, system will change into the right pattern in 20 minutes. Hence, this mode is suitable for a building with variant traffic pattern.

Mode 2: Scheduler catches the output data of neural network from yesterday. The mode is suitable for the building with regular pattern and the patterns in everyday are almost the same.

Mode 3: Scheduler catches the output data of neural network from a week ago. This pattern is modified from mode 2. The system with mode 3 will perform well in office building.

Mode 4: Scheduler catches the output data of neural network from a month ago. Mode 4 is expanded from mode 3. This is for special case to reach the goal of building manager.

4. Simulation

Implementation

The hierarchical view of the elevators system is shown in Fig. 12. In this paper, we focus on the Elevator Schedule Kernel and the interface between schedule kernel and control kernel. Fig. 13, 14, 15 show the simulation result.

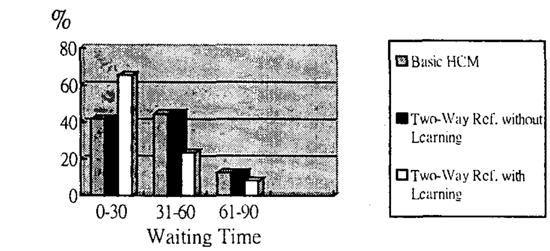

Figure 13: Simulation with Up Peak traffic pattern

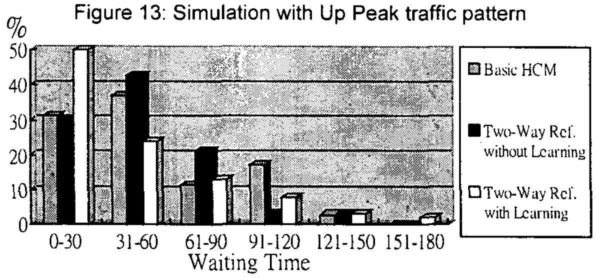

Figure 14: Simulation with Down Peak traffic pattern

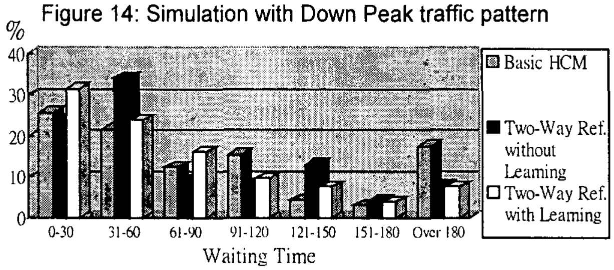

Figure 15: Simulation with Inter Floor traffic pattern

From the simulation result, we can see the effect of the *learning ability*. The learning ability appears to minimize the average waiting time especially in Up Peak traffic pattern. The *Two-Way Refinement Algorithm* seems to reduce the long-wait probability especially in Down Peak traffic pattern. The theoretical model fits the experimental data well.

5. Conclusion

In this paper, we proposed a systematic Petri net based modeling for an elevator system. The entire model is composed of four types of modules, *Basic Movement Module*, *Loading/Unloading Module*, *Direction Reversing Module*, and *Call Management Modules*. In these modules, the control places are utilized to simplify the complexity of the Petri net model, and the control rules are used to satisfy human nature and to approach the goal of efficiency.

To obtain an efficient schedule, we propose an event driven dynamic scheduling and corresponding algorithms. The strategy helps us to handle an elevator system with insufficient information and uncertain events. We estimate the cost of dispatching by Hall Call Assignment Method. The assignment method precisely computes the cost over our CTTPN model. And the *Two-Way Refinement Algorithm* is simple and efficient to refine our schedule in a real time system. The learning ability also helps us to know the traffic pattern the elevator system is running. A historic data base and traffic pattern analyzer has been included to make the learning ability more precisely.

Reference

[1] C. A. Petri, "Kommunikation mit Automaten," Ph.D. Dissertation, University of Bonn, Bonn, West-Germany, 1962.

[2] James L. Peterson, *Petri Net Theory and the Modeling of Systems*, Prentice-Hall, INC., Englewood Cliffs, N.J. 07632, 1981.

[3] Chin-Jung Tasi, Li-Chen Fu and Yung-Jen Hsu, "Modeling and Simulation for Flexible Manufacturing Systems Using Petri Net," *2nd Int. Conf. on Automat.*, vol.4, pp.31-38, 1992.

[4] Chu-Hui Lin and Li-Chen Fu, "Petri Net Based Dynamic Scheduling of an Elevator System," *IEEE Int. Conf. on Robotics and Automat.*, vol.1, pp.192-199, 1996.

[5] Jain, Anil K., and Mao, Jianchang, "Artificial Neural Networks: A Tutorial," *Computer* vol. 29, no. 3, pp. 31-44, Mar., 1996.

[6] Simon Haykin, "Neural Networks (A Comprehensive Foundation)," Macmillan College Publish Company, Inc. 1994.

[7] Sperduti, Alessandro, and Starita, Antonia, "Supervised Neural Networks for the Classification of Structures," *IEEE Trans. on Neural Networks* vol. 8, no. 3, pp. 714-735, May, 1997.

[8] Kogiantis, Achilles G., and Papantoni-Kazakos, Titsa, "Operation and Learing in Neural Networks for Robust Prediction," *IEEE Trans. Syst., Man, and Cybern.*, Part B, vol. 27, no 3, pp. 402-411, 1997.

[9] N. Kameli, K. Thangavelu, "Intelligent Elevator Dispatching System," *AI Expert*, pp. 32-37, Sep. 1989.

[10] H. Aoki, K. Sasaki, "Group Supervisory Control System Assisted by Artificial Intelligence," *Elevator World*, pp. 70-80, Feb. 1990.

[11] Chang Bum Kim, Kyoung A. Seong, Hyung Lee-Kwang, Jeong O. Kim, and Yong Bae Lim, "A Fuzzy Approach to Elevator Group Control System," *IEEE Trans. Syst. Man Cybern.*, vol. 25. NO. 6, pp. 985-990, Jun. 1995.

[12] Chang Bum Kim, Kyoung A. Seong, Hyung Lee-Kwang, and Jeong O. Kim, "Design and Implementation of a Fuzzy Elevator Group Control System," *IEEE Trans. on Syst., Man, and Cybern.*, Part A, vol. 28, no. 3, pp. 277-287, May 1998.

[13] S. Tsuji, M. Amano, S. Hikita, "Application of the Expert System to Elevator Group-Supervisory Control," *IEEE 5th Conf. of Aritifical Intelligence Applications*, pp. 288-294, 1989.

[14] Young Cheol Cho, Zavarin Gagov, and Wook Hyun Kwon, "Elevator Group Control with Accurate Estimation of Hall Call Waiting Times," *IEEE Int. Conf. on Robotics and Automat.*, vol.1, pp. 447-452, 1999.