

Research Article

Sensor and Dynamic Pricing Aware Vertical Transportation in Smart Buildings

Hyunkyung Choi,¹ Kyungwoon Cho,² and Hyokyung Bahn ¹

¹Department of Computer Engineering, Ewha University, Seoul 120-750, Republic of Korea

²Embedded Software Research Center, Ewha University, Seoul 120-750, Republic of Korea

Correspondence should be addressed to Hyokyung Bahn; bahn@ewha.ac.kr

Received 25 December 2018; Revised 13 May 2019; Accepted 27 May 2019; Published 20 June 2019

Academic Editor: Hassan Zargarzadeh

Copyright © 2019 Hyunkyung Choi et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

In modern smart buildings, the electricity consumption of a building is monitored every time and costs differently at each time slot of a day. Smart buildings are also equipped with indoor sensors that can track the movement of human beings. In this paper, we propose a new elevator control system (ECS) that utilizes two kinds of context information in smart buildings: (1) human movements estimated by indoor sensors and (2) dynamic changes of electricity price. In particular, indoor sensors recognize elevator passengers before they press the elevator call buttons, and smart meters inform the dynamically changing price of the electricity to ECS. By using this information, our ECS aims at minimizing both the electricity cost and the waiting time of passengers. As this is a complex optimization problem, we use an evolutionary computation technique based on genetic algorithms (GA). We inject a learning module into the control unit of ECS, which monitors the change of the electricity price and the passengers' traffic detected by sensors. Experimental results with the simulator we developed show that our ECS outperforms the scheduling configuration that does not consider sensor information or electricity price changes.

1. Introduction

With the recent advances in indoor sensor technologies as well as the increasing diffusion of smart meters, various kinds of smart living services within a modern building are being realized. In particular, with the enhancement of sensor technologies, various indoor contexts such as temperature, motion, and light can be instantly detected. Moreover, the electricity consumption of a building can also be detected every time and will cost differently at each time slot of a day.

In this paper, we present a new elevator control system (ECS) that utilizes two kinds of context information in smart green buildings: (1) human movement estimated by indoor sensor technologies and (2) dynamic changes of electricity prices. Firstly, as human activities and movements can be monitored precisely with modern indoor sensors like camera, floor, and optical sensor technologies as shown in Figure 1 [1, 2], elevator control systems can be aware of the passenger's arrival before they press the elevator call buttons. So, we exploit it for the efficient scheduling of

elevator cars, thereby improving passenger's waiting time. Secondly, as the price of electricity dynamically changes in modern smart grid infrastructures, energy consumption at peak hours is more important than other situations [3]. To achieve the maximized benefit under the dynamic pricing system of current smart grid infrastructures, it is necessary to equip a control mechanism that automatically reacts to the change of electricity price without explicit actions from the system admin or tenants. Thus, our elevator control system focuses on the minimization of total electricity charge (i.e., the weighted sum of the energy consumption considering the electricity price of each time slot) rather than total energy consumption within a full day, which was not the main focus of previous studies.

Minimizing both the electricity charge and the passengers' waiting time is not a simple issue as they sometimes conflict. For example, decreasing the number of working elevator cars will be helpful in saving the electricity charge, but it may increase the waiting time of passengers. Furthermore, because of the complex elevator dynamics, complicated



FIGURE 1: Human tracking technologies with indoor sensors [1, 2].

traffic patterns, and multiple objectives to be optimized, state-of-the-art studies use various optimization skills like genetic algorithms and/or fuzzy systems [4–6]. Some studies estimate elevator usage patterns by making use of peak hour distribution analysis [5]. However, estimation and utilization that existing studies use are limited since elevator control systems can be aware of the arrival of passengers only after they press the call buttons. Unlike existing studies, the proposed elevator control system recognizes the arrival of passengers with indoor sensor technologies before they actually press the call buttons, which is then used for efficient scheduling of elevator cars.

With respect to the waiting time criterion, it is important to guarantee the threshold of tolerable waiting time by passengers. In general, passengers’ dissatisfaction grows as their waiting time increases, but there is a certain threshold that passengers are difficult to tolerate [4, 5, 7]. That is, when a passenger’s waiting time is over this threshold, his/her dissatisfaction grows rapidly. Thus, our elevator control system is designed to keep the threshold of tolerable waiting time by passengers even though we need to pay more electricity charge. Our elevator control system is adaptively controlled to support passengers’ satisfaction as well as to minimize the electricity charge. To do so, we inject a learning module into the control unit of the elevator control system, which monitors the change of the electricity price and the passengers’ traffic detected by sensors. By so doing, when the number of passengers becomes large (i.e., heavy traffic), our ECS increases the number of working elevator cars. This will eventually guarantee the threshold of tolerable waiting time by passengers. On the contrary, when the price of electricity becomes high or the elevator traffic lessens, ECS reduces the number of working elevator cars. This will eventually save the electricity charge. However, this should be carefully designed to meet the threshold of tolerable waiting time. Thus, we use a genetic algorithm (GA) to satisfy the two conflicting objectives [8]. Although GA requires a large sequence of operations to decide the scheduling, recent advances in many-core technologies make the real-time decision of elevator scheduling possible in our ECS.

To show the effectiveness of the proposed elevator control system, we conduct experiments under various traffic conditions. Our simulation results show that the proposed ECS outperforms existing elevator systems in terms of the average waiting time, the worst-case waiting time, and the electricity charge. Specifically, the performance gain in terms of the average waiting time and the electricity cost is 5-14% and 58-75%, respectively.

The remainder of this paper is organized as follows. In Section 2, we briefly summarize the state-of-the-art elevator control policies, especially focus on group elevator control. Section 3 describes the proposed elevator control system precisely. Section 4 shows the performance evaluation results to assess the effectiveness of the proposed system. Finally, Section 5 concludes this paper.

2. Literature Review

Group elevator control is a complex optimization problem in a large search space, and thus kinds of optimization skills like fuzzy logics [9–15], genetic network programming [16], DNA computing [17], neural networks [18–20], and genetic algorithms [6, 21–23] have been adopted.

Some studies use branch and bound strategies for pruning the large search space. For example, DNA computing is adopted to prune the large search space [17]. Decision tree is also adopted to limit the complex search space, thereby reducing the computational complexity significantly [24]. Hiller and Klug generate a new candidate schedule set whenever an elevator request arrives and prune the schedules in the set by making use of branch-and-bound techniques [25].

Fuzzy logic is one of the widely used skills for group elevator control [26]. Igarashi et al. propose a group elevator scheduling policy based on fuzzy logic [11]. When an elevator user presses a hall call button, the scheduling policy assesses all candidate cars with the fuzzy function and allocates the car with the minimum cost. Ishikawa et al. [12] and Kaneko et al. [13] also perform group elevator control with the fuzzy logic. In their policy, not only the waiting time but also the riding time of elevator users is considered in determining

the cost of elevator cars. Kim et al. also use fuzzy logic, but they consider energy cost as well as the waiting time of elevator users [9]. The drawback of fuzzy logic based elevator controls above is that it could not undo the scheduling results after the completion of allocations. Fernández et al. solve this problem by adopting an approach that uses dynamic allocation [27]. Siikonen uses fuzzy logic in recognizing the traffic pattern and the traffic peaks from statistical forecasts [15].

Genetic algorithm is another widely used technique for group elevator control. Bolat et al. present a genetic algorithm based elevator control policy that focuses on the riding time of elevator users instead of the waiting time [28]. Ahn et al. also use genetic algorithms for group elevator control but they also consider the electricity cost [4]. Kim and Moon also use genetic algorithms to perform group elevator control. They install cameras at each floor and estimate the number of waiting users, optimizing the group elevator scheduling even more. In particular, they allocate multiple cars to the same floor if the number of waiting users is larger than the remaining capacity of a car. Dai et al. also use genetic algorithms in group elevator control but they assume that the destination floor of a user can be known when a hall call is issued [29]. This is possible if hall call uses destination-floor buttons instead of up-down buttons [30]. Tyni and Ylinen introduce a multiobjective optimization approach based on genetic algorithms in controlling group elevators [23]. Similar to genetic algorithms, a particle swarm algorithm can be adopted in group elevator control [31]. A particle swarm algorithm encodes each solution as a particle, which has an associated speed to move within the solution space, allowing for finding better schedules.

Neural network is also adopted in group elevator control [18, 20]. Neural network has a merit in elevator control as it can easily adapt to sudden traffic changes although their cost is not negligible. Gu et al. propose a hybrid technique that uses both fuzzy logic and neural network. In their policy, fuzzy logic is adopted to manage abrupt traffic changes and neural network is used for considering long-term traffic [32]. Multiagent system is also shown to be effective in fast adaptation to abrupt traffic fluctuations [33–35]. This is because multiagent systems have independent decentralized agents, which allow the quick and robust responses to sudden traffic changes.

Genetic network programming is also considered for group elevator control. One drawback of genetic network programming is that its convergence is very slow and needs heavy computations. To resolve this issue, Mabu et al. incorporate the ant-colony algorithm into genetic network programming, leading to fast convergence to the global optimum [36].

Petri net is also used in group elevator control, but it needs heavy computations for convergence, making it hard to use in real world systems [37]. Tabu search can be an alternative technique for fast convergence in group elevator control [38]. Viral algorithm is also tried for the efficiency of group elevator control and is shown to be better than genetic algorithm or tabu search [39].

Some studies exploit user activities or elevator usage patterns in group-elevator control. In most building environments, elevator usage patterns show periodicity during a day, and certain research groups estimate elevator usage patterns by making use of peak hour distribution analysis [5, 40, 41]. Pepyne and Cassandras present a group elevator control by analyzing up-peak usage patterns of elevator systems [41, 42]. Brand et al. make use of both down-peak and up-peak usage patterns in their elevator control [40]. Their system allocates an elevator car to a parking location before real passengers come. The parking location is decided by analyzing down-peak and up-peak usage patterns. Note that parking schemes can reduce the waiting time of elevator users when the parking was performed at an appropriate floor.

The usage pattern of an elevator evolves as time progresses. Some research has been made to find and model such changes. Dewen et al. propose a learning scheme for catching the usage pattern changes [18]. Their proposed scheme detects the current usage patterns and chooses a control method between a certain set of scheduling policies. Pepyne et al. propose an elevator control policy that postpones the movement of cars when the number of passengers is below a certain threshold [41]. Zhang and Zong observe that the energy consumption of an elevator car becomes large in up-peak cases and propose a low-power control for up-peak cases [43].

As estimation of usage patterns can be incorrect, some studies recognize elevator users before they arrive by making use of some additional facilities. RFID and camera are used together to recognize the arrival of elevator users and their target floors before pressing call buttons, which is used for intelligent elevator control [44, 45]. Some studies change the control algorithm itself by the detecting result of RFID [46].

3. The Proposed System

In this section, we describe the details of the proposed system called EPSEC (Electricity Price and Sensor aware Elevator Control). EPSEC is different from conventional elevator scheduling systems in that it adaptively controls elevator scheduling considering the dynamic change of electricity price and the passenger traffic monitored by indoor sensor technologies, thereby minimizing the electricity cost and waiting time. To do so, we inject a learning module into the control unit of EPSEC, which monitors the change of the electricity price and the passengers' traffic detected by sensors.

Figure 2 depicts the overall structure of EPSEC; it is composed of three units, namely, the cost evaluation unit, the assignment unit, and the control unit. The cost evaluation unit evaluates the estimated cost of each elevator car to serve a request. The assignment unit determines the car to serve the request. The control unit keeps track of the price of the electricity and the passenger traffic with indoor sensors and periodically adjusts the control parameter.

In the elevator request interfaces, there are two types of call buttons: *hall call* and *car call* [9, 47]. A hall call is triggered when a passenger presses upward or downward button at hall in order to ride an elevator car. A car call is triggered

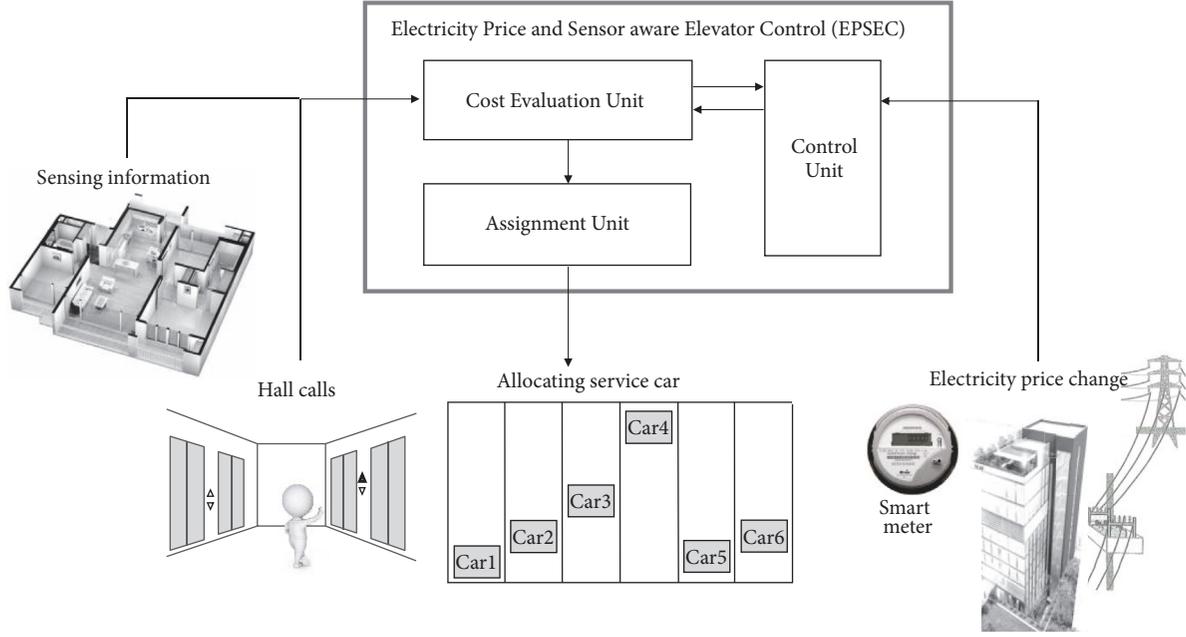


FIGURE 2: Basic architecture of EPSEC.

when a passenger rides a car and presses the destination-floor button in the elevator car. As we use sensor technologies to estimate candidate passengers, there is another kind of call, a *reservation call*. A reservation call is issued when indoor sensors detect the arrival of candidate passengers and estimate the time when the passenger will reach the elevator and generate a hall call based on the estimated time.

When a hall call occurs, the elevator control system determines the car to serve that request. In our system, the cost evaluation unit calculates the estimated cost of each elevator car from its current location to the requested floor. We represent the cost as the weighted sum of the electricity charge and the expected waiting time. The expected waiting time $COST_{TIME}$ of a car w from the m -th floor to the n -th floor is computed as

$$COST_{TIME}(w, m, n) = \sum (TIME_{MOVE} + TIME_{STOP}) \quad (1)$$

$$TIME_{MOVE}$$

$$= \begin{cases} 2 * \sqrt{\frac{DIST}{ASC}} & \text{if } DIST < DIST_{MIN} \\ \frac{DIST}{ASC} + \frac{ASC}{USC} & \text{otherwise} \end{cases} \quad (2)$$

$$TIME_{STOP}$$

$$= TIME_{OPEN} + TIME_{BOARD} + TIME_{CLOSE} \quad (3)$$

where $DIST$ is the moving distance of the car from its current position to the next stop position, $DIST_{MIN}$ is the minimum distance required for constant speed motion, ASC is the constant acceleration of the car, and USC is the uniform velocity of the car. $TIME_{OPEN}$, $TIME_{BOARD}$, and $TIME_{CLOSE}$

represent the latency to open the elevator door, board the car, and close the elevator door, respectively. We set constant values for $TIME_{OPEN}$, $TIME_{BOARD}$, and $TIME_{CLOSE}$. If the car needs to visit multiple floors while moving from the m -th floor to the n -th floor, each $TIME_{MOVE}$ and $TIME_{STOP}$ are added to $COST_{TIME}$.

Besides velocity and acceleration, the rate of change of acceleration, which we call *jerk*, needs to be modeled and limited in elevator kinematics, for passengers to withstand without discomfort [48–50]. Unfortunately, we could not incorporate the formulation of jerks into our simulator, for now, and thus our results do not reflect the effect of jerk. However, as we only focus on the scheduling algorithm itself rather than the precise modeling of passenger's waiting time, our modeling without considering jerk does not seem to affect the effectiveness of the algorithm.

The electricity charge $COST_{CHARGE}$ of the car w from the m -th floor to the n -th floor is determined based on the movement of an elevator car as follows.

$$COST_{CHARGE}(w, m, n) = \left\{ \sum (E_{ACCEL} + E_{UNI}) + E_{WAKEUP} \right\} C_{PRICE} \quad (4)$$

$$E_{UNI} = |W_{CAR} + W_P - W_{CWT}| gh \quad (5)$$

where E_{ACCEL} and E_{UNI} are energy consumptions in accelerating and constant speed motions, respectively, E_{WAKEUP} is the wakeup energy from the hibernation state, and C_{PRICE} is the electricity price at that time. W_{CAR} is the mass of the car, W_P is the sum of passengers' weight, W_{CWT} is the weight of the counterweight, g is the gravity constant, and h is the distance between the current floor and the target floor. We set E_{ACCEL} as a constant value as accelerating an elevator car is a short

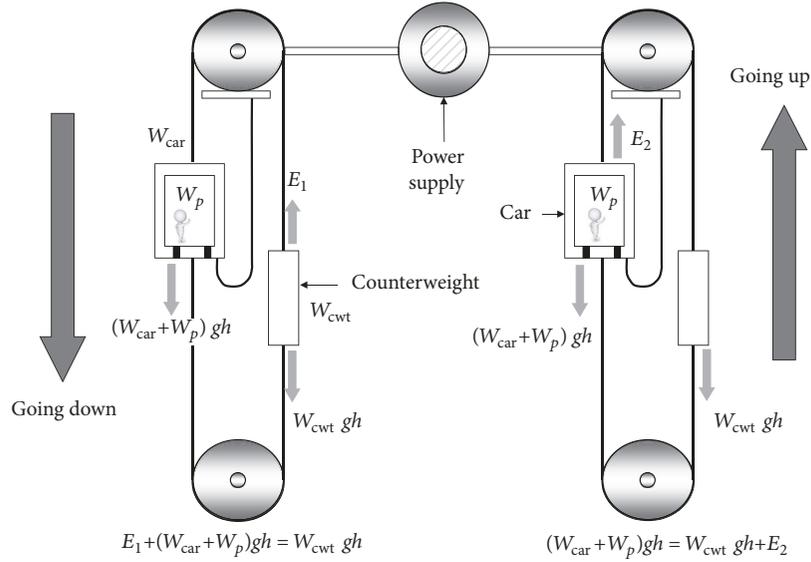


FIGURE 3: Energy principle of elevator movements.

motion, which happens within a short period of 0.1 to 0.5 seconds [7]. If a car needs to visit multiple floors during its movement, each E_{ACCEL} is accumulated. Figure 3 shows the energy principle of elevator movements.

After calculating all cars' cost with respect to the waiting time and the electricity charge, the cost evaluation unit sends the results to the assignment unit. Then, the assignment unit allocates the car that is expected to incur the minimum cost to this request. The total cost function $\Phi(w, m, n)$ of a car w from the m -th floor to the n -th floor is evaluated by the weighted sum of $COST_{TIME}(w, m, n)$ and $COST_{CHARGE}(w, m, n)$ as follows.

$$\Phi(w, m, n) = \alpha COST_{TIME}(w, m, n) + (1-\alpha) COST_{CHARGE}(w, m, n) \quad (6)$$

where α ($0 \leq \alpha \leq 1$) is a weight parameter that decides how much emphasis is given to either the waiting time factor or the electricity cost factor. The car with the least total cost is determined as the service car.

There are various ways to consider two heterogeneous objectives with a single unified cost function. As the relative importance of these two objectives (i.e., waiting time and electricity charge) in this paper is varied according to situations such as user traffic and electricity price, we use a weighted sum and adjust the relative weight of the two objectives as the situation changes. Although the numerical scale of the two cost functions may not be commensurate, it can be calibrated by setting a weight parameter α appropriately.

The cost evaluation unit delivers the waiting time and the electricity cost information to the control unit. Based on this information, the control unit monitors the state of the system and periodically updates the control parameter α . The initial value of the parameter α is set by the system admin and is updated periodically based on the average waiting time of the latest interval in order to keep the passenger's

tolerable waiting time. The threshold of the tolerable waiting time is set to the maximum round trip time (RTT) of an elevator car divided by the number of elevator cars [7]. When the passenger traffic becomes very high, the control unit increases the value of α to keep the threshold of tolerable waiting time although it is not the admin's intention. When the traffic returns to the normal state, the control unit decreases the value of α to save the electricity charge. Even though the passenger traffic becomes very light, the system sets α to the initial value and does not decrease any more to conform to the admin's intention.

The control parameter α is also adjusted when the electricity price changes. In particular, when the price of electricity becomes high, the system decreases α to save the electricity charge. However, this is not a simple problem as decreasing α may do harm to the tolerable waiting time of passengers.

To resolve this issue, we use an optimization technique based on genetic algorithms. Genetic algorithm (GA) is a probabilistic algorithm which behaves similar to the natural evolution of population genetics in problem solving or simulation [8]. In our problem, tuning the control parameter α considering the two conflicting objectives is not a simple issue and thus we use GAs. Given the passengers' request history in the previous time interval, GA determines the new scheduling algorithm of the next time interval by determining an appropriate α value. To do this, GA maintains certain candidate solutions and evolves the solution set to find the best solution that determines the scheduling algorithm of the next time interval. Specifically, GA first generates an initial solution set and two solutions are selected from the set and then merged as one solution. This is performed by the two major GA operations: crossover and mutation. Then, the solution set is evolved by substituting the newly generated solution for a solution in the old set. This process is continued until the solution set converges. Then, the best solution in the

final set is selected to generate the scheduling algorithm of the next time interval.

3.1. Encoding and Cost Functions. In a GA, a solution is typically represented by a linear string. As our problem is a numerical optimization problem, floating-point encoding is used in our GA design [51]. That is, each position in a string has a value from 0 to 9 and the position of decimal point is not fixed. The initial set is composed of 100 random solutions. In a GA, a cost function is necessary to measure the value of each solution. As our goal is minimizing the electricity charge without missing the tolerable waiting time of passengers, we define our cost function as the electricity charge that would be yielded by the scheduling algorithm the solution represents; and we add some penalty cost to a solution if the scheduling algorithm does not meet the threshold of tolerable waiting time by passengers.

3.2. Selection Operations. In the selection operation, two solutions in the current set are selected, which will be the parents of the next generation. In this paper, we normalize the fitness of the solutions so that the best one in the set is 4 times more probable to be chosen than the worst one. This is a standard normalization method frequently used in GAs [52]. The normalized fitness of a solution i is defined as

$$Norm_i = (Rank_w - Rank_i) + \frac{(Rank_w - Rank_b)}{3} \quad (7)$$

where $Rank_w$, $Rank_b$, and $Rank_i$ are the rankings of the worst solution, the best solution, and solution i , respectively, in the current set. With this normalized fitness, we use the roulette wheel selection operation, in which each solution i is assigned a slot with the size equal to its normalized fitness $Norm_i$. A random number ranging from 0 to $\sum Norm_i$ is generated and the pointer of the wheel indicates that position. Then, a solution whose slot is indicated by the wheel pointer is chosen and becomes the parent solutions. Note that this is based on a stochastic rule that favors superior solutions but is not deterministic. We use a probabilistic approach here to avoid too much discrimination. That is, if we deterministically choose the best solutions, they can dominate the solution set quickly, leading to premature convergence to a local optimum.

3.3. Crossover Operations. After selecting two parent solutions, our GA performs a crossover operation. The rationale behind crossover is that superior parts of a parent need to be survived to make a child, which will generate better offspring as time progresses. The most widely used crossover operator is the 1-point crossover, which works by randomly selecting a cut-point within the solution and then a child is generated by copying the left part of a parent and the right part of the other parent. Instead of using classical 1-point crossovers, we use an arithmetic crossover operation as our problem is a numerical optimization problem, in which classical crossovers may not be effective to transfer the parents' properties to a child [51].

Specifically, in our algorithm, a child is generated by the weighted sum of the two parent solutions as follows.

$$Child = r Parent_1 + (1 - r) Parent_2, \quad 0 \leq r \leq 1 \quad (8)$$

It combines two parent solutions from a numerical viewpoint; it is known to be more natural for numerical optimization problems. Arithmetic crossover may reduce the population diversity and consequently cause premature convergence to a poor local optimum. We can compensate this problem by mutation.

3.4. Mutation Operations. Mutation perturbs some parts of a child solution generated by crossover. It helps search wide area of problem space and avoids premature convergence. A classical mutation operator selects some random part of a solution and changes it to a random value. However, classical mutation may cause inappropriate effects in case of numerical optimization problems. In a floating-point encoding, if the selected part for mutation is the most significant position, the effect of mutation is too great. On the contrary, if the selected part is the least significant position that is below the decimal point, the effect of mutation may be negligible. Thus, we use a nonuniform mutation, in which less significant positions of the solution get higher probability of being mutated as time progresses, while more significant positions get lower probability. That is, such a mutation allows the global search of the problem domain in the early stage of the GA; local search is strengthened later. Nonuniform mutation has the ability of fine tuning which helps fast convergence of GAs [53].

3.5. Replacement Operations. When a child solution is generated, a new solution set is produced by substituting the generated child with a solution in the current generation. This paper replaces the solution that incurs the highest cost in the current set by the newly generated one, which is the most commonly used replacement operation.

3.6. Overhead of Our GA. Elevator scheduling is a typical combinational optimization problem and GA is a well-known technique to efficiently assign and schedule elevator calls to cars. This paper also uses GA in elevator scheduling, but instead of using GA in finding the actual schedule of each elevator call, our GA determines the scheduling algorithm itself by tuning the control parameter α according as the passenger traffic and the electricity price change. In GA, each solution is evaluated based on a cost function, and a solution in our case determines the scheduling algorithm. Thus, to evaluate a solution, scheduling should be virtually performed for a series of elevator calls during a certain time period. Thus, the overhead of our GA in evaluating each solution is relatively heavier than usual combinatorial optimization problems. However, our GA is triggered periodically and determines a new scheduling algorithm for the next epoch by setting an appropriate α value given the passengers' request history of the previous epoch. Note that the default epoch of our experiment is set to 10 minutes and the average number of calls within this epoch is 50 to 500. As our problem is a kind

TABLE 1: Simulation configurations.

Parameter	Value
# of floors	20
# of cars	6
Capacity of a car	20 people/car
Uniform velocity	3 m/sec
Acceleration	1 m/sec ²
Height of a floor	3 m
Open and close latency	2 sec
Boarding latency	1.5 sec /passenger

of function optimization, we use numerical encoding and arithmetic crossover operations instead of binary encoding and cut-point based crossover operations usually adopted in combinatorial optimizations.

For fast convergence, we model our problem as a single objective optimization with a constraint instead of multiobjective optimization techniques. In particular, the problem is modeled as the minimization of the electricity charge with a constraint that the waiting time is below a certain threshold. This is reasonable because guaranteeing the tolerable waiting time is important but passengers can endure if the waiting time is less than a certain threshold. This allows the fast convergence of our GA within at most 5,000 repetitions. Due to the advances in modern embedded many-core technologies, such computation overhead is within acceptable ranges.

4. Performance Evaluation

4.1. Experimental Setup. We conduct simulation experiments to assess the effectiveness of the proposed scheduling system. Specifically, our simulations were performed under two workload conditions. They consist of homogeneous traffic and nonhomogeneous traffic. Table 1 lists the experimental configurations used in our experiments. The number of floors is 20 and the number of cars is 6, respectively. The maximum capacity of an elevator car is 20 passengers. In homogenous traffic, the traffic of passengers is generated by the Poisson process with the average arrival rates of 5 to 50 passengers/minute similar to previous studies [6]. In nonhomogenous traffic, we consider dynamically changing traffic patterns during a day. Elevator traffic of a typical office building relies on if a major flow is an upward direction from the main entrance (up-peak), a downward direction toward the main entrance (down-peak), their mixture (up-peak/down-peak), or none of them (inter-floor) [54]. In particular, the up-peak happens at the beginning of the working day, the down-peak at the end of the working day, and their mixture at the lunch time. Inter-floor happens when passengers randomly move between floors during the daytime and there is no peak traffic at this time. Our non-homogeneous traffic includes up-peak (8:00-9:00), down-peak (17:00-18:00), up/down-peak (12:00-13:00), and inter-floor patterns as is done in previous studies [9, 47, 54, 55]. The arrival rate for the up-peak and down-peak is set to 30 passengers/minute and 15 passengers/minute, respectively.

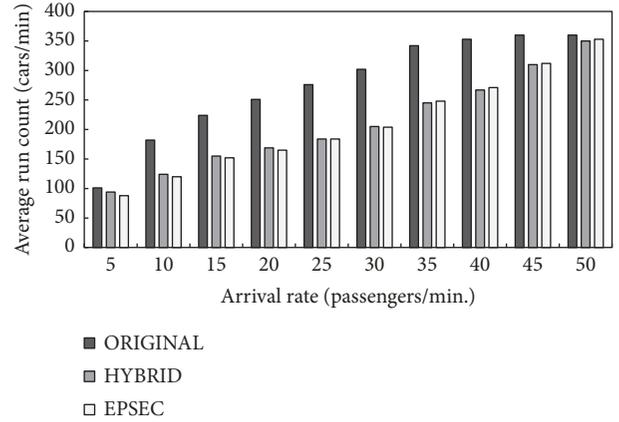


FIGURE 4: Average run count of cars as a function of the arrival rate.

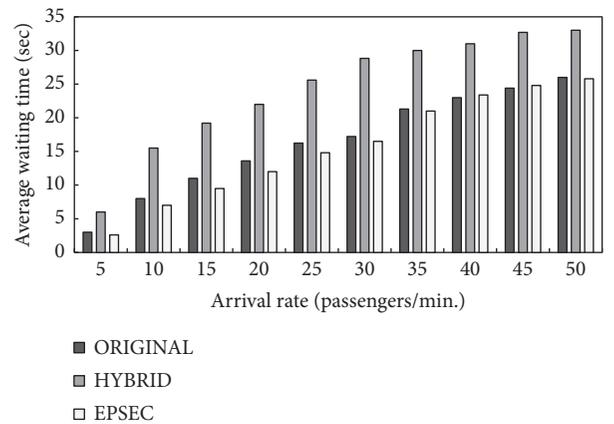


FIGURE 5: Passenger's waiting time as a function of the arrival rate.

4.2. Experimental Results. We compare our elevator scheduling system (EPSEC) with ORIGINAL, HYBRID, and Hill Climbing. ORIGINAL is a usual elevator system that does not consider either the electricity price or the sensor information. In our experiment, the cost function of the ORIGINAL is set to the waiting time of the passenger as defined in Expression (1). HYBRID takes into account both passengers' waiting time and the electricity price but does not consider the tolerable waiting time by passengers. That is, HYBRID uses the same cost function of EPSEC given in Expression (6) but fixes the parameter α to the initial value set by the system admin without adjustment. Also, HYBRID does not use optimization techniques like genetic algorithms. Hill Climbing uses the same cost function of EPSEC and also utilizes the sensor information. That is, Hill Climbing considers electricity price changes, sensor information, and the tolerable waiting time by passengers similar to EPSEC. The only difference between Hill Climbing and EPSEC lies in the tuning method of the parameter α . In particular, Hill Climbing adjusts the control parameter α by making use of the simple hill climbing search method.

Figures 4 and 5 show the results for the homogeneous traffic. Figure 4 depicts the average run count of elevator cars as the arrival rate of passengers is varied. As shown in

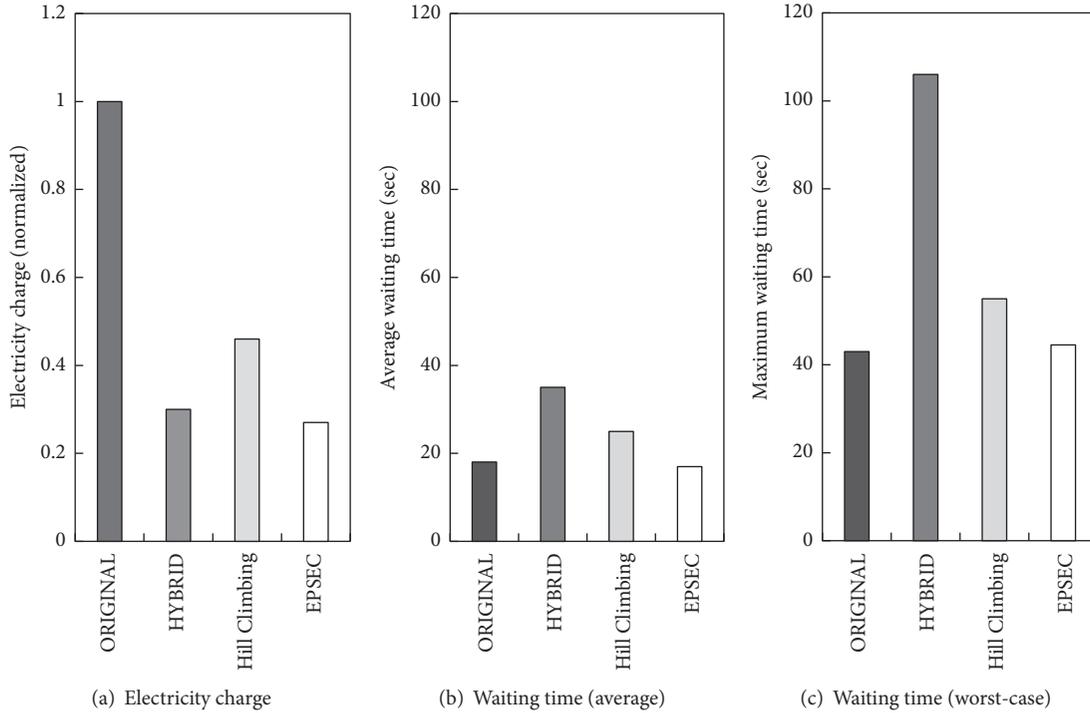


FIGURE 6: Results with nonhomogeneous workload conditions.

the figure, ORIGINAL performs the worst in terms of the average run count as it does not take into account the energy factor. HYBRID and EPSEC commonly reduce the average run count significantly. Though the results of HYBRID and EPSEC are similar, HYBRID does not have the functionality of supporting the threshold of the tolerable waiting time by passengers.

Figure 5 depicts the average waiting time with the homogeneous traffic as the arrival rate is varied. Though EPSEC considers the electricity charge as well as the average waiting time, in most cases, it performs better than ORIGINAL that only takes into account the average waiting time. Unlike EPSEC, the average waiting time of HYBRID degrades seriously when the arrival rate is more than 30 (passengers/minute). The reason is that HYBRID does not take into account the tolerable waiting time although the passenger traffic increases. EPSEC performs relatively worse than ORIGINAL when the arrival rate is 45 (passengers/minute), but it is not a serious issue as the average waiting time is within the tolerable threshold under this condition. Actually, EPSEC saves much electricity charge in such situations.

When we analyze our elevator configurations listed in Table 1 based on the typical elevator traffic model [48], the up-peak handling capacity is about 35 persons/minute. However, in our experiments, we cannot see any rapid performance degradations although the arrival rate becomes over 35 persons/minute. We could not find any particular reason for this phenomenon, but the reason we conjecture is that the traffic used in this experiment is a homogeneous traffic that does not have peak time traffic.

To see the effectiveness of the proposed scheduling system under more realistic situations, we simulate the case of nonhomogeneous traffic with the mixture of up-peak (8:00-9:00), down-peak (17:00-18:00), up/down-peak (12:00-13:00), and inter-floor patterns. Figure 6 shows the electricity charge, the average waiting time, and the worst-case waiting time of the four schemes when nonhomogeneous traffic is applied. As can be seen from the figures, EPSEC shows competitive performance with respect to both the electricity charge and the average waiting time. In particular, with respect to the electricity charge, EPSEC performs the best for all cases. The average reduction in the electricity charge in comparison with ORIGINAL is 72%. It performs even better than HYBRID that does not consider the threshold of the tolerable waiting time by passengers. Figures 6(b) and 6(c) show the average and the worst-case waiting time of the four schemes. As shown in the figure, ORIGINAL and EPSEC exhibit similar results. In contrast, HYBRID performs significantly worse than the other two schemes because it does not consider the workload traffic changes.

Now, let us discuss the results of Hill Climbing. As shown in Figure 6(a), the electricity charge of Hill Climbing is significantly higher than that of EPSEC. This is because EPSEC maintains solutions not meeting the threshold of tolerable waiting time at a certain level by giving some penalty cost while searching the solution space, whereas Hill Climbing prunes the solution space that does not meet the threshold of tolerable waiting time. Nevertheless, the waiting time of Hill Climbing is worse than EPSEC in terms of both the average and the worst cases as shown in Figures 6(b)

and 6(c). This is because Hill Climbing remains at a local optimum and is not able to search better solution spaces any longer. Note that the performance improvement of EPSEC against Hill Climbing is 41%, 32%, and 19%, respectively, in terms of the electricity charge, the average waiting time, and the worst-case waiting time.

4.3. Sensor Information. The time delay between the sensor detecting a passenger and that passenger registering the actual call depends on the passenger's moving time as well as the location of sensors. Unlike elevator's moving time that can be precisely estimated by the law of motion, passenger's moving time varies depending on the distance of the sensor from the elevator and the passenger's step. If we overestimate the passenger's moving time, the passenger should wait long for the elevator car. If we underestimate it, the elevator car may arrive before the passenger arrives. To precisely estimate passenger's moving time, we use multiple sensor devices. Specifically, we adopt 3 sensors at each floor and locate them at 10-, 20-, and 30-second walking distance from the elevator's position. As a closer sensor estimates the moving time of the candidate passenger more precisely, a reservation call from a distant sensor is dropped from the request queue when it appears from a closer sensor. To inform the scheduler of an accurate time delay between the reservation call and the actual call, passenger's moving time can be precisely estimated by calculating the time difference detected by the two adjacent sensors.

To assess the effectiveness of reservation calls with indoor sensors, we made a simple prototype with multiple sensor devices. Specifically, we equipped 3 types of sensors at a floor of a building and collected sensed data during a day. We then compared the reservation calls issued by sensor information and actual hall calls from passengers. We used a CCTV in front of the elevator door to collect the actual hall call information. The comparison result showed that 85.2% of the reservation calls eventually invoke actual calls. As reservation calls are dropped from the request queue if subsequent actual calls do not appear, the effect of false reservation calls is not significant. Note that our scheduler waits only for a certain limited time for reservation calls, and it promptly reacts to the changed request information. In contrast, most requests benefit from these reservation calls by shortening their waiting time.

5. Conclusions

As smart grid and indoor sensor technologies are advancing, various smart living services are being realized. This paper proposed a new elevator control system that utilizes two kinds of context information in smart green buildings: human movement estimated by indoor sensor technologies and dynamic changes of electricity price. Firstly, human activities and movements can be precisely monitored with indoor sensor technologies, and thus our elevator control system recognizes the arrival of passengers before they press the actual call buttons. Then, we exploit it for the scheduling efficiency of elevator cars, thereby improving passenger's waiting time. Secondly, as the price of electricity dynamically

changes in modern smart grid infrastructures, our elevator scheduling system focuses on the minimization of total electricity charge.

Our elevator scheduling algorithm takes into account the electricity price changes during a day and the tolerable waiting time by passengers with sensor technologies. In particular, the objective of our algorithm is to minimize the electricity charge without increasing passengers' dissatisfaction. To do this, we model and solve the problem with genetic algorithms. To validate the efficiency of the proposed system, we conducted experiments under various workload configurations including homogeneous and nonhomogeneous traffic workloads. Experimental results with the simulator we developed showed that the proposed elevator control system reduces the electricity charge of the elevators by 72% on average compared to our simulator configuration that does not consider sensor information or electricity price changes. Nevertheless, it reduces the average waiting time and the worst-case waiting time of passengers.

Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this article.

References

- [1] I. A. Essa, "Ubiquitous sensing for smart and aware environments," *IEEE Personal Communications*, vol. 7, no. 5, pp. 47–49, 2000.
- [2] R. J. Orr and G. D. Abowd, "The smart floor: a mechanism for natural user identification and tracking," in *Proceedings of the Int. Conf. Human Factors in Computing Systems*, pp. 275–276, 2000.
- [3] T. T. Kim and H. V. Poor, "Scheduling power consumption with price uncertainty," *IEEE Transactions on Smart Grid*, vol. 2, no. 3, pp. 519–527, 2011.
- [4] S. Ahn, S. Lee, and H. Bahn, "A smart elevator scheduler that considers dynamic changes of energy cost and user traffic," *Integrated Computer-Aided Engineering*, vol. 24, no. 2, pp. 187–202, 2017.
- [5] M. Brand and D. Nikovski, "Optimal parking in group elevator control," in *Proceedings of the IEEE International Conference on Robotics & Automation*, pp. 1002–1008, 2004.
- [6] J.-H. Kim and B.-R. Moon, "Adaptive elevator group control with camera," *IEEE Transactions on Industrial Electronics*, vol. 48, no. 2, pp. 377–382, 2001.
- [7] K. S. Koh, *Introduction to Elevator System*, Seiwunsa, 1990.
- [8] M. Mitchell, *An Introduction to Genetic Algorithms*, MIT Press, Cambridge, Mass, USA, 1996.
- [9] C. Kim, K. A. Seong, H. Lee-Kwang, and J. O. Kim, "Design and implementation of a fuzzy elevator group control system," *IEEE Transactions on Systems, Man and Cybernetics, Part A: Systems and Humans*, vol. 28, no. 3, pp. 277–287, 1998.

- [10] R. Gudwin, F. Gomide, and M. A. Netto, "A fuzzy elevator group controller with linear context adaptation," in *Proceedings of the IEEE International Conference on Fuzzy Systems*, pp. 481–486, 1998.
- [11] K. Igarashi, S. Take, and T. Ishikawa, "Supervisory control for elevator group with fuzzy expert system," in *Proceedings of the IEEE International Conference on Industrial Technology*, pp. 133–137, 1994.
- [12] T. Ishikawa, A. Miyauchi, and M. Kaneko, "Supervisory control for elevator group by using fuzzy expert system which also addresses traveling time," in *Proceedings of the IEEE International Conference on Industrial Technology*, pp. 87–94, 2000.
- [13] M. Kaneko, T. Ishikawa, and Y. Sogawa, "Supervisory control for elevator group by using fuzzy expert system," in *Proceedings of the 23rd International Conference on Industrial Electronics, Control, and Instrumentation*, pp. 370–376, 1997.
- [14] K. Kurosawa and K. Hirasawa, "Intelligent and supervisory control for elevator group," *Transactions on Information Processing Society of Japan*, vol. 26, pp. 278–287, 1985.
- [15] M. L. Siikonen, *Elevator Group Control with Artificial Intelligence*, University of Technology, Helsinki, Finland, 1997.
- [16] T. Eguchi, K. Hirasawa, H. Jinglu, and S. Markon, "Elevator group supervisory control systems using genetic network programming," *Congress on Evolutionary Computation*, vol. 2, pp. 1661–1667, 2004.
- [17] H. Zhao and X. Liu, "An improved DNA computing method for elevator scheduling problem," *Lecture Notes in Computer Science*, vol. 7719, pp. 869–875, 2013.
- [18] Z. Dewen, J. Li, Z. Yuwen, S. Guanghui, and H. Kai, "Modern elevator group supervisory control systems and neural networks technique," in *Proceedings of the IEEE International Conference on Intelligent Processing Systems*, pp. 528–532, 1997.
- [19] S. Markon, H. Kita, and Y. Nishikawa, "Adaptive optimal elevator group control by use of neural network," *Transactions on Institute of Systems, Control and Information Engineers*, vol. 7, pp. 487–497, 1994.
- [20] A. So, J. Beebe, W. Chan, and S. Liu, "Elevator traffic pattern recognition by artificial neural network," *Elevator Technology*, vol. 6, pp. 122–131, 1995.
- [21] A. Fujino, T. Tobita, K. Segawa, K. Yoneda, and A. Togawa, "An elevator group control system with floor attribute control method and system optimization using genetic algorithms," in *Proceedings of the 21st International Conference on Industrial Electronics, Control, and Instrumentation*, pp. 1502–1507, 1995.
- [22] A. Fujino, T. Tobita, K. Segawa, K. Yoneda, and A. Togawa, "An elevator group control system with floor-attribute system optimization using genetic algorithms," *IEEE Transactions on Industrial Electronics*, vol. 44, pp. 546–552, 1997.
- [23] T. Tyni and J. Ylinen, "Evolutionary bi-objective optimisation in the elevator car routing problem," *European Journal of Operational Research*, vol. 169, no. 3, pp. 960–977, 2006.
- [24] M. Hamdi and D. J. Mulvaney, "Prioritised A* search in real-time elevator dispatching," *Control Engineering Practice*, vol. 15, no. 2, pp. 219–230, 2007.
- [25] B. Hiller, T. Klug, and A. Tuchscherer, "An exact reoptimization algorithm for the scheduling of elevator groups," *Flexible Services and Manufacturing Journal*, vol. 26, pp. 1–23, 2013.
- [26] J. R. Fernandez and P. Cortes, "A survey of elevator group control systems for vertical transportation: a look at recent literature," *IEEE Control Systems Magazine*, vol. 35, no. 4, pp. 38–55, 2015.
- [27] J. Fernández, P. Cortés, J. Munuzuri, and J. Guadix, "Dynamic fuzzy logic elevator group control system with relative waiting time consideration," *IEEE Transactions on Industrial Electronics*, vol. 61, no. 9, pp. 4912–4919, 2014.
- [28] B. Bolat, P. Cortés, E. Yalçın, and M. Alişverişi, "Optimal car dispatching for elevator groups using genetic algorithms," *Intelligent Automation & Soft Computing*, vol. 16, no. 1, pp. 89–99, 2010.
- [29] D. Dai, J. Zhang, W. Xie, Z. Yin, and Y. Zhang, "Elevator group-control policy with destination registration based on hybrid genetic algorithms," in *Proceedings of the IEEE International Conference on Computer Application and System Modeling*, pp. 535–538, 2010.
- [30] M. Amano, M. Yamazaki, and H. Ikejima, "The latest elevator group supervisory control system," *Elevator Technology*, vol. 6, pp. 88–95, 1995.
- [31] B. Bolat, O. Altun, and P. Cortés, "A particle swarm optimization algorithm for optimal car-call allocation in elevator group control systems," *Applied Soft Computing*, vol. 13, no. 5, pp. 2633–2642, 2012.
- [32] C. Gu, G. Zhao, and Z. Liu, "Adaptive fuzzy logical elevator group control system based on artificial neural net," *Computer Measurement & Control*, vol. 11, no. 1, pp. 947–949, 2003.
- [33] D. Muñoz, C. Llanos, M. Ayala-Rincón, R. van Els, and R. Almeida, "Implementation of dispatching algorithms for elevator systems using reconfigurable architectures," *Engineering Applications of Artificial Intelligence*, vol. 21, no. 8, pp. 1309–1320, 2008.
- [34] Y. Ogoshi, H. Kimura, S. Hirose, and N. Osato, "Elevator group control system using multiagent system," *Systems and Computers in Japan*, vol. 34, no. 1, pp. 45–58, 2003.
- [35] Z. T. Wang, D. F. Ji, F. Qiao, and Q. D. Wu, "Modeling and scheduling elevator group control system based on MAS and negotiation mechanism of agents," *Control and Decision*, vol. 22, no. 10, pp. 1184–1188, 2007.
- [36] S. Mabu, L. Yu, J. Zhou, S. Eto, and K. Hirasawa, "A double-decker elevator systems controller with idle cage assignment algorithm using genetic network programming," *Journal of Advanced Computational Intelligence and Intelligent Informatics*, vol. 114, no. 5, pp. 487–496, 2010.
- [37] Z. Qun, S. Ding, C. Yu, and L. Xiaofeng, "Elevator group control system modeling based on object oriented petri net," *Elevator World*, vol. 49, no. 8, pp. 99–105, 2001.
- [38] B. Bolat and P. Cortés, "Genetic and tabu search approaches for optimizing the hall call - Car allocation problem in elevator group systems," *Applied Soft Computing*, vol. 11, no. 2, pp. 1792–1800, 2011.
- [39] P. Cortes, L. Onieva, J. Munuzuri, and J. Guadix, "A viral system algorithm to optimize the car dispatching in elevator group control systems of tall buildings," *Computers & Industrial Engineering*, vol. 64, no. 1, pp. 403–411, 2013.
- [40] M. Brand and D. Nikovski, "Risk-averse group elevator scheduling," in *Proceedings of the Elevcon World Congress on Vertical Transportation Technology*, pp. 1–8, 2004.
- [41] D. L. Pepyne and C. G. Cassandras, "Optimal dispatching control for elevator systems during uppeak traffic," *IEEE Transactions on Control Systems Technology*, vol. 5, no. 6, pp. 629–643, 1997.
- [42] D. L. Pepyne and C. G. Cassandras, "Design and implementation of an adaptive dispatching controller for elevator systems during uppeak traffic," *IEEE Transactions on Control Systems Technology*, vol. 6, no. 5, pp. 635–650, 1998.

- [43] J. Zhang and Q. Zong, "Energy-saving scheduling optimization under up-peak traffic for group elevator system in building," *Energy and Buildings*, vol. 66, pp. 495–504, 2013.
- [44] O. Kwon, E. Lee, and H. Bahn, "Sensor-aware elevator scheduling for smart building environments," *Building and Environment*, vol. 72, pp. 332–342, 2014.
- [45] N. Shimbun, *Elevators Get Smart*, RFID in Japan, 2004.
- [46] T. Strang and C. Bauer, "Context-aware elevator scheduling," in *Proceedings of the International Conference on Advanced Information Networking and Applications Workshops*, pp. 276–281, 2007.
- [47] A. Fujino, T. Tobita, and K. Yoneda, "An on-line tuning method for multi-objective control of elevator group," in *Proceedings of the IEEE Industrial Electronics, Control, Instrumentation, and Automation*, pp. 795–800, 1992.
- [48] G. Barney, *Elevator Traffic Handbook*, Spon Press, 2003.
- [49] N. R. Roschier and M. Kaakinen, "New formulae for elevator round trip calculation," Supplement to Elevator World for ACIST Members, pp 189–197, 1979.
- [50] R. Peters, *Vertical transportation planning in buildings [Ph.D. thesis]*, Brunel University, 1998.
- [51] H. Kim and H. Adeli, "Discrete cost optimization of composite floors using a floating-point genetic algorithm," *Engineering Optimization*, vol. 33, no. 4, pp. 485–501, 2001.
- [52] D. Goldberg, *Genetic Algorithms in Search, Optimization, and Machine Learning*, Addison-Wesley, Boston, Mass, USA, 1989.
- [53] Z. Michalewicz, *Genetic Algorithms + Data Structures = Evolutionary Programs*, Springer, Heidelberg, Germany, 3rd edition, 1996.
- [54] P. Cortés, J. Muñuzuri, and L. Onieva, "Design and analysis of a tool for planning and simulating dynamic vertical transport," *Simulation: Transactions of the Society for Modelling and Simulation International*, vol. 82, no. 4, pp. 255–274, 2006.
- [55] J. Fernández, P. Cortés, J. Guadix, and J. Muñuzuri, "Dynamic fuzzy logic elevator group control system for energy optimization," *International Journal of Information Technology & Decision Making*, vol. 12, no. 3, pp. 591–617, 2013.

