

O.R. Applications

Dynamic optimization of the operation of single-car elevator systems with destination hall call registration: Part I. Formulation and simulations

Shunji Tanaka^{*}, Yukihiro Uraguchi, Mituhiko Araki¹

Graduate School of Electrical Engineering, Kyoto University, Kyotodaigaku-Katsura, Nishikyo-ku, Kyoto 615-8510, Japan

Received 29 November 2002; accepted 18 April 2004

Available online 4 July 2004

Abstract

The purpose of this two-part study is to investigate the operation problem of single-car elevator systems with destination hall call registration. Destination hall call registration is such a system in which passengers register their destination floors at elevator halls before boarding the car, while in the ordinary systems passengers specify only the directions of their destination floors at elevator halls and register destination floors after boarding the car. In this part of the study, we formulate the operation problem as a dynamic optimization problem and demonstrate by computer simulations that dynamically optimized operation considerably improves the transportation capability compared to conventional selective collective operation. How to solve the dynamic optimization problem is given in the second part of this study.

© 2004 Elsevier B.V. All rights reserved.

Keywords: Transportation; Elevator operation; Destination hall call registration; Dynamic optimization; Selective collective operation

1. Introduction

Along with the development of high rise buildings, high performance is becoming strongly required for elevator systems. The most straightforward way of improving elevator performance is to install larger or additional elevator cars or to accelerate the speed of elevator cars. However, such hardware approaches

^{*} Corresponding author. Tel.: +81 75 383 2204; fax: +81 75 383 2201.

E-mail addresses: tanaka@kuee.kyoto-u.ac.jp (S. Tanaka), uraguchi@rotary.kuee.kyoto-u.ac.jp (Y. Uraguchi), araki@kuee.kyoto-u.ac.jp (M. Araki).

¹ Tel.: +81 75 383 2200; fax: +81 75 383 2201.

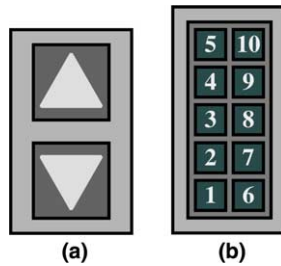


Fig. 1. Two types of hall call buttons: (a) the ordinary system and (b) the destination hall call registration system.

are often restricted by space limitation, cost problems, safety requirements, and other conditions. Hence, intelligent elevator control has been studied so far by many researchers [1–9].

As a solution to intelligent elevator control, the destination hall call registration system [10] has taken notice recently. In the destination hall call registration system passengers register their destination floors at elevator halls before boarding the cars, while in the ordinary systems passengers specify only the directions of their destination floors at elevator halls (see Fig. 1) and register the destination floors after boarding the cars.

Today there are several types of elevator control systems with destination hall call registration. These control systems utilize the information about destination floors of passengers for passenger dispatching (allocation of passengers to individual elevator cars) so that each car's stops are reduced and, as a result, overall elevator performance is improved. For example, assume that an elevator system consisting of two cars is installed and four passengers P_1 , P_2 , P_3 and P_4 are waiting at floor 1 whose destination floors are 5, 5, 6 and 6, respectively. In the ordinary systems, their destination floors are not known when the elevator controller dispatches the passengers to the cars. Therefore, it can happen that P_1 and P_3 are dispatched to one car, and P_2 and P_4 to the other car. In this case, both the cars should stop at floors 5 and 6. On the other hand, in the destination hall call registration system the elevator controller can dispatch the passengers to the cars so that each car's stops are reduced: P_1 and P_2 are dispatched to one car, and P_3 and P_4 to the other. Since each car stops only once in this allocation (the former car stops at floor 5 and the latter at floor 6), overall elevator performance is improved.

However, these elevator controllers do not make full use of the information about destination floors. In such elevator control systems it is used only for passenger dispatching and each car's operation to serve dispatched passengers is determined by (single-car) selective collective operation [10] that is widely used in the existing elevator control systems. This operation only requires the information about the directions of passengers waiting at elevator halls, and does not make use of the information about their destination floors until they board the car. Thus, it is expected that the utilization of destination information for each car's operation improves elevator performance more.

In this study we examine how to determine car's operation by utilizing destination information and whether operation performance can be improved. For this purpose, we consider the operation problem of *single-car* elevator systems with destination hall call registration. We formulate this problem as a dynamic optimization problem such that operation is re-optimized at every instant when a new hall call is registered (a passenger registers his/her destination floor at an elevator hall). Then, we demonstrate by computer simulations that dynamically optimized operation considerably improves the transportation capability compared to conventional selective collective operation when passenger traffic is heavy.

In the context of intelligent elevator control, there are only few studies on the operation problem itself although many researchers have studied passenger dispatching. Levy et al. [5] treated the operation problem as a stochastic optimization problem and showed some simple "optimal policies". However, it is hard

to apply their method to more realistic elevator operation problems because they only gave simple formulae that cannot be solved without several approximations. As an alternative, we employ a deterministic approach. It has the disadvantage that future arrival of passengers cannot be taken into account, but in return gives an explicit solution algorithm that can be applied to realistic elevator operation problems.

Recently, Inamoto et al. [9] tackled the static elevator control problem for multi-car elevator systems where all the information about passengers is assumed to be ideally known a priori, and proposed a branch-and-bound algorithm to optimize passenger dispatching and car operation at once. However, their problem setting is too simplified; the load/unload time is independent of the number of passengers loaded or unloaded, the car travels at a constant speed, and so on. Moreover, their algorithm can handle problem instances with at most 20 passengers even in the case of single-car elevator systems, and thus is too slow to be applied to dynamic operation optimization even for simulation purposes. In this study we will treat a more realistic elevator operation problem, and propose a branch-and-bound algorithm for dynamic operation optimization, which is fast enough at least for simulation-based analysis.

This study consists of two parts. In Part I, we formulate the elevator operation problem of single-car elevator systems with destination hall call registration as a dynamic optimization problem, and examine efficiency of dynamically optimized operation by computer simulations. In Part II, a branch-and-bound algorithm to solve the dynamic optimization problem is constructed.

The outline of Part I is as follows. In Section 2, we state specifications and assumptions necessary for the formulation of the elevator operation problem. In Section 3, we formulate the operation problem as a dynamic optimization problem. In Section 4, we compare by computer simulations dynamically optimized operation with selective collective operation, which is widely used in the existing elevator systems. Last, the obtained results are summarized in Section 5.

2. Specifications and assumptions on the elevator system

In this section we describe explicit specifications and assumptions necessary for problem formulation.

Consider an M -storey building ($M=10$) in which a single-car elevator system is installed. This elevator system has destination hall call buttons at every elevator hall that can specify passengers' destination floors (see Fig. 1(b)). Every passenger is supposed to register his/her destination floor (hall call) by pushing a button even if his/her destination floor has already been registered by another passenger who arrived at the hall earlier. A guide system is also installed at every elevator hall to inform passengers of when they should board the car. This guide system is destination-oriented: It indicates at each hall the destination floors of passengers who should board the car when the car stops there next.

The load capacity of the car, N_C , is given by the maximum number of passengers that the car can load. The physical performance of the car is given by Table 1, which is typical for elevators installed in 10-storey or 15-storey buildings. In Table 1(a), the floor-to-floor traveling time stands for the time required for the car to travel from one floor to another. The deceleration time in Table 1(b) stands for the time for the car to stop at a floor. The door-closing time and door-opening time are given in Table 1(c). The load/unload time in Table 1(d) stands for the time for the car to load or unload one passenger.

For example, consider the case that the car stays at floor 1 with the door closed and will serve two passengers who are going to travel from floor 1 to floor 10. For the passengers to board the car, the door starts opening at 0.00 second and finishes opening at $0.00 + 2.20 = 2.20$ seconds (Table 1(c)). Then, the first passenger starts boarding the car, and finishes boarding at $2.20 + 0.80 = 3.00$ seconds (Table 1(d)). The second passenger finishes boarding at $3.00 + 0.80 = 3.80$ seconds. The door starts closing immediately and finishes closing at $3.80 + 2.20 = 6.00$ seconds. Then, the car starts acceleration and travels toward floor 10. Since the car should travel nine floors, it takes 19.17 seconds to reach floor 10 (Table 1(a)). For this 9-floor-travel, the deceleration time is 4.57 seconds (Table 1(b)) and thus the car starts deceleration at

Table 1
Performance of the car

Floors	Time (seconds)
<i>(a) Floor-to-floor traveling time</i>	
1	6.09
2	8.04
3	9.57
4	11.17
5	12.77
6	14.37
7	15.97
8	17.57
9	19.17
<i>(b) Deceleration time</i>	
1	2.92
2	3.90
3 or more	4.57
<i>(c) Door-closing and door-opening times</i>	
Door-closing time	2.20
Door-opening time	2.20
<i>(d) Load/unload time</i>	
Load/unload time	0.80 seconds/1 person

$6.00 + 19.17 - 4.57 = 20.60$ seconds. The car continues decelerating and stops at floor 10 at $20.60 + 4.57 = 25.17$ seconds. For the passengers to leave the car, the door starts opening and finishes opening at $25.17 + 2.20 = 27.37$ seconds. The first passenger starts leaving the car, and finishes leaving at $27.37 + 0.80 = 28.17$ seconds. Finally, the second passenger starts leaving, and finishes leaving at $28.17 + 0.80 = 28.97$ seconds.

On this elevator system, we make several assumptions. First, we make three assumptions to simplify the problem.

Assumption 2.1. If the door starts opening or closing, it cannot be interrupted. Similarly, if a passenger starts boarding or leaving the car, it cannot be interrupted.

Assumption 2.2. The car always keeps track of the number of on-board passengers. When the number of on-board passengers reaches the load capacity, the car immediately closes the door, starts traveling, and does not stop at any floor without unloading a passenger.

Assumption 2.3. If there are no passengers to be served, the car stays at the last visited floor with the door closed.

In some elevator systems, the (empty) car returns to the main floor (floor 1) and stays there with the door opened after all passengers have been served. Of course it is not difficult to treat such elevator operation, but in this study we restrict our discussions to the elevator operation satisfying Assumption 2.3.

The following two assumptions concerning elevator operation arise from the mechanical limitations of elevator systems.

Assumption 2.4. The car cannot stop or reverse halfway between floors.

Assumption 2.5. If the car starts deceleration, it cannot accelerate again until it stops (at a floor).

On the other hand, the following assumptions are necessary for, so to speak, natural elevator operation.

Assumption 2.6. On-board passengers cannot leave the car at any floors other than their destination floors.

Assumption 2.7. The car cannot stop at a floor without loading or unloading any passengers there.

In real elevator systems, Assumption 2.7 is often violated because of passengers' unexpected behavior such as pushing wrong buttons, leaving elevator halls without boarding the car, and so on. However, in this study we assume that all passengers do not make mistakes and are *honest*, so that Assumption 2.7 is satisfied. Please note that this assumption also excludes some kind of elevator operation that does not occur in real elevator systems. For example, let us consider the case that a new passenger arrives at floor 4 when the empty car is moving upward around floor 5 to serve a passenger waiting at floor 10. Without Assumption 2.7, the car can stop at floor 6 (or 7) without loading any passengers, reverse its direction, and serve the passenger waiting at floor 4 before the one waiting at floor 10. Clearly, such operation does not occur in real elevator systems.

The following assumptions are also necessary for natural elevator operation.

Assumption 2.8. Passengers cannot board the car before all on-board passengers who are going to leave the car there at that stop finish leaving the car.

Assumption 2.9. If there are two or more passengers who are going to board the car at a floor, they should board the car in the ascending order of the arrival times (hall call registration times).

Assumption 2.10. If there are two or more passengers who are going to leave the car at a floor, they should leave the car in the ascending order of the arrival times.

Assumption 2.10 does not reflect practical situations since the order of passengers' leaving the car depends, in general, on their standing positions in the car. However, since we cannot simulate such passengers' behavior completely, we should assume that the order is determined by some rule.

The following two assumptions are to forbid psychologically undesirable operations for passengers.

Assumption 2.11. If the car loads a passenger at a floor, it must load the other passengers at that stop who are waiting there and have the same destination floors with him/her as many as possible (to the maximum load capacity).

Assumption 2.12. When the car stops at a floor, it must unload all the on-board passengers at that stop whose destination floors are there.

Assumption 2.12 forbids such an operation that on-board passengers cannot leave the car even when the car stops at their destination floors. On the other hand, Assumption 2.11 forbids such an operation that when the car stops at a floor where passengers having the same destination floor are waiting, only some of them can board the car even if the car can afford.

Assumption 2.11 is also necessary from the implementation viewpoint of elevator systems. In the ordinary elevator systems, when passengers should board the car is clear: it is when the car travel direction and their desired direction are the same (and, of course, the car can afford to load them). In this study, however, we consider such an operation in that when passengers should board the car is not determined only by the car travel direction. Therefore, a guide system to indicate when passengers should board the car at each floor is necessary for realization of our elevator operation, whether or not Assumption 2.11 is made.

If Assumption 2.11 is not made, we should install such a guide system that is not user-friendly and thus is unrealistic. For example, consider that two passengers are waiting at floor 1 whose destination floors are floor 5. Without Assumption 2.11, it is possible that the car does not load the passengers at once; the car loads one of the passengers, travels to other floors, returns to floor 1, and then loads the other passenger. In this case, the guide system must inform them separately of when they should board the car. It could be realized, for example, by numbering passengers individually and offering information by the numbers. But it is not user-friendly and is unrealistic. If Assumption 2.11 is made, we only need a destination-oriented guide system at each floor that indicates the destination floors of passengers who should board the car when the car stops there next.

The following assumption is concerning the special operation called *reversal* [5].

Assumption 2.13. While at least one passenger is on-board, the car cannot reverse its direction.

This assumption excludes such an operation in that some passenger travels in the opposite direction of his/her destination floor after boarding the car. This operation, called reversal, is forbidden in principle since it is psychologically undesirable for passengers [5]. Nevertheless, reversal has a potential to improve the efficiency of elevator operation especially when destination hall call registration is used. Therefore, in this study we consider two types of problems; the problems with and without this assumption (with reversal forbidden and permitted, respectively).

Here, we should note that Assumption 2.12 is always satisfied if Assumption 2.13 is satisfied. If Assumption 2.12 is not satisfied, it follows that the car passes through some passenger's destination floor while he/she is on-board, and, as a result, reversal occurs. Thus, Assumption 2.12 is redundant when Assumption 2.13 is made.

3. Formulation of the elevator operation problem

In this section we formulate the elevator operation problem under the specifications and assumptions given in the preceding section.

As mentioned in Section 1, we formulate the elevator operation problem as a dynamic optimization problem such that elevator operation is re-optimized at every instant when a new hall call is registered. However, a difficulty arises if we re-optimize elevator operation while a passenger is boarding the car.

Let us consider, for example, the case that one of three passengers waiting at floor 1 (their destination floors are different) starts boarding the car at 0 second and a new hall call is registered at 0.7 seconds. Consider further that the operation schedule before 0.7 seconds is such that all the three passengers board the car at that stop. Since a new hall call is registered, we are to re-optimize the operation so as not to violate Assumptions 2.1–2.12 (and Assumption 2.13). According to Assumptions 2.1 and 2.7, at least one passenger must board the car at that stop (at floor 1) in the re-optimized operation. In other words, the assumptions are satisfied even if only one or two passengers board the car at that stop.

Now, assume that the re-optimized operation is such that only the first passenger boards the car and the second and third passengers do not at that stop. Since these passengers were scheduled to board the car in the operation before the re-optimization, they are urged to board the car by the guide system until 0.7 seconds. The new hall call is registered at 0.7 seconds and the operation is modified so that they do not board the car at that stop. Thus, the guide system should stop urging them at 0.7 seconds. But it is too late for the second passenger, who was scheduled to board the car at 0.8 seconds in the original operation, to stop boarding because only 0.1 seconds is left for him/her.

To avoid such a difficulty, we make the following additional assumption.

Assumption 3.1. If the car starts deceleration to stop at a floor, the schedule of passengers to board or leave the car there at that stop cannot be modified.

Under this assumption in addition to Assumptions 2.1–2.12 (and Assumption 2.13), we are to formulate the elevator operation problem as a dynamic optimization problem. More specifically, we formulate the optimization problem to be solved at the instant t_S when a new hall call is registered.

Let us define the set of on-board passengers at t_S by

$$\mathcal{B} = \{P_k \mid P_k \text{ is on-board at } t_S\} \quad (1)$$

and the set of waiting passengers at t_S (passengers who have registered their hall calls before t_S , but do not board the car yet at t_S) by

$$\mathcal{W} = \{P_l \mid P_l \text{ is waiting for the car at } t_S\}. \quad (2)$$

Then, our problem to be formulated here is to optimize the elevator operation that serves \mathcal{B} and \mathcal{W} from t_S . In the following, we will first clarify the objective of elevator control and determine the objective function of this problem. Next, we will shift t_S to reduce the possible car states at t_S so that the formulation becomes easy. Finally, we will give our formulation.

3.1. The performance index of elevator operation and the objective function of the dynamic optimization problem

The primary objective of elevator control is to improve service quality for passengers. For this reason, performance indices based on passengers' waiting times have been commonly used for the evaluation of elevator operation. Here, the waiting time of a passenger is defined by the time from the instant when he/she registers his/her destination floor (the arrival time) until the instant when he/she boards the car.

Clearly, it is better for elevator operation to reduce passengers' waiting times as much as possible, but it is not sufficient for our problem: If reversal is permitted (without Assumption 2.13), such an unrealistic operation becomes "good operation" that the car loads waiting passengers as many as possible without unloading any on-board passengers. Therefore, we adopt the average and maximum service times instead as performance indices for elevator operation. The service time of a passenger stands for the time from the instant when he/she registers his/her destination floor until the instant when he/she leaves the car at his/her destination floor.

It is another issue what type of objective function we should adopt for the dynamic optimization problem. The objective function should be such that the average and maximum service times are as small as possible in the overall operation obtained by dynamic operation optimization. However, it cannot be realized by simply choosing the average service time as the objective function. For example, let us consider a passenger whose arrival time is 5 seconds in the operation that minimizes the average service time at 10 seconds ($t_S = 10$). Since the change of the average service time caused by the change of passengers' arrival times is constant, this operation remains optimal even if the arrival time of the passenger is changed to 0 or 10 seconds. This implies that individual arrival times are not taken into account in this objective function and thus passengers who arrived at elevator halls earlier cannot always board the car earlier. Therefore, as pointed out in [11], it is possible that the service for a particular passenger is deferred every time when operation is re-optimized and, as a result, the maximum service time deteriorates much. To avoid such a situation, we adopt the weighted average service time as the objective function of the dynamic optimization problem. By putting larger weights on passengers with earlier arrival times, the maximum service time is expected to be reduced. How to adjust the weights will be examined in Section 4.

3.2. Reduction of car states

Our problem to be formulated in this section is to find the elevator operation from t_S that minimizes the weighted average service time of the passengers belonging to $\mathcal{B} \cup \mathcal{W}$. Thus, the car states at t_S correspond to the initial car states for the optimization problem. In order to make the formulation easy, we reduce these car states before the formulation.

There are following six possible car states at t_S .

- State 1.* The car stays at a floor with the door closed.
- State 2.* The car stays at a floor and the door is closing.
- State 3.* The car stays at a floor and the door is opening.
- State 4.* The car stays at a floor and a passenger is boarding or leaving the car.
- State 5.* The car is decelerating.
- State 6.* The car is accelerating or moving at a constant speed.

Here, the car is assumed to be in the latter state if it is just in between two states. For example, the car state when the door finishes opening (just between State 3 and State 4) is assumed to be State 4.

Since we should take account of these states as the initial car states in the problem formulation, it is better to reduce the states as much as possible. In the following, we reduce the car states by shifting t_S .

First, we consider State 2. From Assumption 2.1, elevator operation cannot be modified until the time t'_S when the door finishes closing. It follows that we only need to optimize elevator operation after t'_S . Thus, in this case we redefine t_S by t'_S and assume that the car state at t_S is State 1.

Next, we consider States 3–5. From Assumption 3.1, once the car starts decelerating to stop at a floor, elevator operation cannot be modified until all the passengers that were scheduled to board or leave the car at that stop finish boarding or leaving. Moreover, Assumption 2.11 requires that if the newly arrived passenger at t_S has the same origin and destination floors with some passenger who was scheduled to board the car at that stop, this newly arrived passenger should also board the car at that stop (if the car can afford). It follows that we only need to optimize elevator operation after t''_S when all these passengers finish boarding or leaving. Thus, in this case we redefine t_S by t''_S and assume that the car state at t_S is

State 7. The car stays at a floor with the door opened.

We also redefine \mathcal{W} and \mathcal{B} by

$$\mathcal{W} := \mathcal{W} - \mathcal{W}_{\text{board}}, \quad (3)$$

$$\mathcal{B} := (\mathcal{B} - \mathcal{B}_{\text{leave}}) \cup \mathcal{W}_{\text{board}}, \quad (4)$$

where $\mathcal{W}_{\text{board}}$ and $\mathcal{B}_{\text{leave}}$, respectively, denote the sets of passengers who board and leave the car at that stop. The passenger who registers his/her hall call at t_S is included in $\mathcal{W}_{\text{board}}$ if necessary.

Last, we consider State 6. Floors at which the car can stop immediately after t_S depend on the position and the speed of the car at t_S , since the car needs deceleration time before it stops. For this reason, we introduce \mathcal{S} , which denotes the set of the floors where the car can stop immediately after t_S . Keeping \mathcal{S} unchanged, we can shift t_S back to t'''_S when the car left the last visited floor. In this case, the initial car state is converted to State 1 (the car stays at the last visited floor with the door closed).

To summarize, we can convert the optimization problem at t_S as if the initial car state is State 1 or 7 (the car stays at a floor with the door closed or opened), by shifting t_S , modifying \mathcal{W} and \mathcal{B} accordingly, and introducing the additional set \mathcal{S} . Fig. 2 and Table 2 describe this conversion.

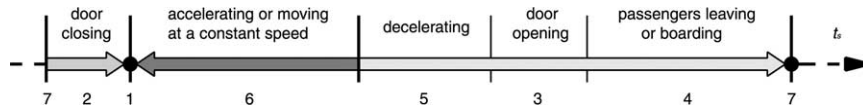
Fig. 2. Shifting t_S .

Table 2

Converted states, t_S , \mathcal{W} , \mathcal{B} and \mathcal{S}

Original car states	State 2	States 3–5	State 6
t_S after shift	t'_S	t''_S	t'''_S
Car states after shift	State 1	State 7	State 1
\mathcal{W} after shift	\mathcal{W}	$\mathcal{W} - \mathcal{W}^{\text{board}}$	\mathcal{W}
\mathcal{B} after shift	\mathcal{B}	$(\mathcal{B} - \mathcal{B}^{\text{leave}}) \cup \mathcal{W}^{\text{board}}$	\mathcal{B}
\mathcal{S} after shift	–	–	\mathcal{S}

For the convenience hereafter, we set $\mathcal{S} = \{1, 2, \dots, M\}$ if the car state before the conversion is not State 6.

3.3. Formulation as a dynamic optimization problem

Let us define $n_B = |\mathcal{B}|$ and $n_W = |\mathcal{W}|$, and denote the passengers belonging to \mathcal{B} and \mathcal{W} by

$$\mathcal{B} = \{P_{b_1}, P_{b_2}, \dots, P_{b_{n_B}}\}, \quad (5)$$

$$\mathcal{W} = \{P_{w_1}, P_{w_2}, \dots, P_{w_{n_W}}\}. \quad (6)$$

For each passenger $P_i \in \mathcal{B} \cup \mathcal{W}$,

- t_i the arrival time,
- o_i the origin floor,
- d_i the destination floor,
- μ_i the weight in the objective function,

are given. Please note that the information about the destination floors of the waiting passengers (d_i for $P_i \in \mathcal{W}$) is not available in the ordinary elevator systems. This point is the primary advantage of destination hall call registration.

Let us define n by $n = n_B + 2n_W$ and e_i ($1 \leq i \leq n$) by

$$e_i = \begin{cases} o_{w_i} & (1 \leq i \leq n_W, P_{w_i} \in \mathcal{W}), \\ d_{w_{i-n_W}} & (n_W + 1 \leq i \leq 2n_W, P_{w_{i-n_W}} \in \mathcal{W}), \\ d_{b_{i-2n_W}} & (2n_W + 1 \leq i \leq n, P_{b_{i-2n_W}} \in \mathcal{B}). \end{cases} \quad (7)$$

From Assumptions 2.1–2.7, the elevator operation problem that we should consider here is to determine the visiting order of e_i so that the weighted average service time is minimized. Therefore, we introduce the decision variables x_i ($1 \leq i \leq n$) defined by

- x_i the i th visiting floor of the car is e_{x_i} .

We also introduce the following variables and constants:

e_0 the initial car position, i.e., the floor where the car stays at t_S ,
 D the initial door state,

$$D = \begin{cases} 0 & \text{the door is closed at } t_S \text{ (the initial car state is State 1),} \\ 1 & \text{The door is opened at } t_S \text{ (the initial car state is State 7).} \end{cases}$$

$T_M(j,i)$ the traveling time from floor i to floor j given by Table 1(a),
 T_C the door-closing time (=2.20),
 T_O the door-opening time (=2.20),
 T_L the load/unload time (=0.80).

Furthermore, we introduce the auxiliary variables T_i ($1 \leq i \leq n$) and B_i ($1 \leq i \leq n$) given by the following definitions:

T_i the time when the car finishes loading or unloading a passenger at e_i ,
 B_i the number of on-board passengers after loading or unloading a passenger at e_i ($B_0 = n_B$).

Then, the problem to find the elevator operation under Assumption 2.13 (with reversal forbidden) that minimizes the weighted average service time is formulated as follows:

$$\min F_P = \frac{1}{n_B + n_W} \left(\sum_{i=1}^{n_W} \mu_{w_i} (T_{i+n_W} - t_{w_i}) + \sum_{i=1}^{n_B} \mu_{b_i} (T_{i+2n_W} - t_{b_i}) \right) \quad (8)$$

$$\text{s.t. } x_i \in \{1, 2, \dots, n\} \quad (1 \leq i \leq n), \quad (9)$$

$$x_i \neq x_j \quad (1 \leq i, j \leq n, i \neq j), \quad (10)$$

$$T_{x_1} = t_S + T_M(e_{x_1}, e_0) + D \cdot T_C + T_O + T_L \quad (e_{x_1} \neq e_0), \quad (11)$$

$$T_{x_1} = t_S + (1 - D) \cdot T_O + T_L \quad (e_{x_1} = e_0), \quad (12)$$

$$T_{x_i} = T_{x_{i-1}} + T_M(e_{x_i}, e_{x_{i-1}}) + T_C + T_O + T_L \quad (2 \leq i \leq n, e_{x_i} \neq e_{x_{i-1}}), \quad (13)$$

$$T_{x_i} = T_{x_{i-1}} + T_L \quad (2 \leq i \leq n, e_{x_i} = e_{x_{i-1}}), \quad (14)$$

$$T_{n_W+i} \geq T_i + T_M(e_{n_W+i}, e_i) + T_C + T_O + T_L \quad (1 \leq i \leq n_W), \quad (15)$$

$$e_{x_1} \in \mathcal{S}, \quad (16)$$

$$B_{x_i} = B_{x_{i-1}} + 1 \quad (1 \leq i \leq n, 1 \leq x_i \leq n_W), \quad (17)$$

$$B_{x_i} = B_{x_{i-1}} - 1 \quad (1 \leq i \leq n, n_W + 1 \leq x_i \leq n), \quad (18)$$

$$B_i \leq N_C \quad (1 \leq i \leq n), \quad (19)$$

$$x_i (1 \leq i \leq n) \text{ satisfy Assumption 2.8,} \quad (20)$$

$$x_i (1 \leq i \leq n) \text{ satisfy Assumption 2.9,} \quad (21)$$

$$x_i (1 \leq i \leq n) \text{ satisfy Assumption 2.10,} \quad (22)$$

$$x_i (1 \leq i \leq n) \text{ satisfy Assumption 2.11,} \quad (23)$$

$$x_i (1 \leq i \leq n) \text{ satisfy Assumption 2.13.} \quad (24)$$

The precedence constraints that the car should visit passengers' origin floors before their destination floors appear in (15). The constraints arising from the car load capacity are (17)–(19).

In the case that Assumption 2.13 is not made (reversal is permitted), we replace (24) by the following constraint:

$$x_i (1 \leq i \leq n) \text{ satisfy Assumption 2.12.} \quad (25)$$

As mentioned in Section 2, this constraint is redundant and thus removed when Assumption 2.13 is made.

Hereafter, we denote the problem (8)–(24) (the problem with reversal forbidden) by (P1), and the problem (8)–(23) and (25) (the problem with reversal permitted) by (P2).

Remark 3.1. Although we can formulate our problem via 0–1 decision variables as standard mixed integer problems, we employ this formulation for the convenience of constructing a solution algorithm, which is given in the second part of this study.

Remark 3.2. Since the terms $\mu_{b_i} t_{b_i}$ and $\mu_{w_i} t_{w_i}$ in (8) are constants, (P1) and (P2) are equivalent to the problems of minimizing

$$\sum_{i=1}^{n_W} \mu_{w_i} T_{i+n_W} + \sum_{i=1}^{n_B} \mu_{b_i} T_{i+2n_W}. \quad (26)$$

Remark 3.3. The optimal solutions of (P1) and (P2) do not change even if we relax the constraint (20). With regard to (22), a weaker condition holds: We can relax (22) in (P1) and (P2) without changing the optimal objective function values if the weights μ_i ($P_i \in \mathcal{B} \cup \mathcal{W}$) are chosen in accordance with t_i , that is,

$$\mu_j \leq \mu_k \quad \text{if} \quad t_j > t_k \quad (27)$$

holds.

The detailed proofs are given in Appendix A.

4. Computer simulations

To solve the dynamic operation optimization problem formulated in the preceding section, we constructed a branch-and-bound algorithm, the details of which are presented in the second part of this study. In this section, we compare by computer simulations dynamically optimized operations with selective collective operation that is widely used in the existing elevator systems (see Appendix B for details). We examine the stationary performance with changing the arrival rate of passengers, the traffic pattern, the car load capacity and the weights for passengers.

To evaluate the stationary performance, we generate the passengers' data by the following procedure. The empty car is assumed to stay at floor 1 with the door closed at 0 second. Then, for a given arrival rate of passengers N_P (person/hour), $N_P/2$ passengers shall arrive from 0 to 1800 seconds (from 0 to 0.5 hour), N_P passengers from 1800 to 5400 seconds (from 0.5 to 1.5 hours), and $N_P/2$ passengers from 5400 to 7200 seconds (from 1.5 to 2 hours). Elevator operation is calculated to serve all the $2N_P$ passengers, but only the N_P passengers who arrive from 1800 to 5400 seconds are taken into consideration to evaluate the stationary performance of operation.

The arrival times of these $2N_P$ passengers are generated by uniform distributions in the corresponding intervals. The origin and destination floors of the passengers are randomly generated according to the following two typical traffic patterns.

1. *Uppeak traffic*. $N_1:N_2:N_3:N_4=19:1:1:1$.
2. *Downpeak traffic*. $N_1:N_2:N_3:N_4=6:13:1:1$.

Here, N_1 , N_2 , N_3 , and N_4 , respectively, denote the number of passengers from floor 1 to floors $[2, M]$, from floors $[2, M]$ to floor 1, from floors $[2, M]$ to floors $[2, M]$ (upward) and from floors $[2, M]$ to floors $[2, M]$ (downward), where M is fixed for 10.

For each combination of N_P and a traffic pattern, 10 sets of passenger data are generated. Then, for each data set the following three types of operations are computed with the load capacity N_C being varied.

- SC selective collective operation,
- D1 dynamically optimized operation with reversal forbidden (corresponds to (P1)),
- D2 dynamically optimized operation with reversal permitted (corresponds to (P2)).

The simulation program, coded by C, is run on a personal computer (Pentium4 2.4GHz).

4.1. Unweighted case

First, we set all the weights for the service times of the passengers to be 1 ($\mu_i = 1$) in (8). In other words, we dynamically optimize the unweighted average service time. The simulation results are shown in Figs. 3 and 4, where the maximum and average service times are evaluated over all the 10 data sets. From Fig. 3 we see that dynamically optimized operation with reversal permitted (D2) is better than that with reversal forbidden (D1), and D1 is better than selective collective operation (SC) with regard to the average service time. Moreover, D1 and D2 outperform SC especially when passenger traffic is heavy (N_P is large) and the load capacity of the car N_C is small. Indeed, Fig. 3(a)(iii) shows that in the case of $N_C = 10$, $N_P \geq 330$ makes the average service time diverge since it exceeds the transportation capability of SC, while D1 and D2 can serve passengers well even when $N_P \geq 330$. These results also show that D1 or D2 for the car with $N_C = 10$ is even better than SC for the car with $N_C = 15$ when passenger traffic is heavy ($N_P \geq 400$). Thus, dynamically optimized operation would enable us to reduce the car capacity (size) necessary for achieving the desired transportation capability compared to selective collective operation.

The reason why dynamically optimized operation possesses better transportation capability than selective collective operation can be explained as follows. First, let us consider the case of uppeak traffic. During uppeak traffic, almost all the passengers arrive at floor 1. However, the number of passengers that the car can load at floor 1 is at most N_C , the car load capacity. Therefore, if passenger traffic is so heavy that the number of newly arrived passengers at floor 1 in one round trip time (the period that the car returns to floor 1) exceeds N_C , the number of waiting passengers increases and, as a consequence, the average service time diverges. In selective collective operation, the car should stop at many floors and round trip times tend to be large because the car loads passengers simply in the order of their arrival times. On the other hand, in dynamically optimized operation the car tries to load passengers with the same destination floor at once. This reduces car stops and makes round trip times small. Since the car returns to floor 1 faster than in selective collective operation, it can serve passengers even when the passenger arrival rate becomes larger.

Examples of car diagrams in SC and D1 during uppeak traffic are shown in Fig. 5, where $N_P = 960$ and $N_C = 10$. In these figures, black and white circles denote car stops for loading and unloading, respectively. We can verify that by reducing the number of car stops for unloading (white circles) in each round trip, dynamically optimized operation makes round trip times smaller than selective collective operation even when the car travels to the highest floor (floor 10).

Next, we consider the case of downpeak traffic. In selective collective operation, the car loads passengers to the maximum load capacity while traveling downwards from the highest floor when passenger traffic becomes heavier. In such a case, the car continues to be full until it reaches floor 1 because almost all

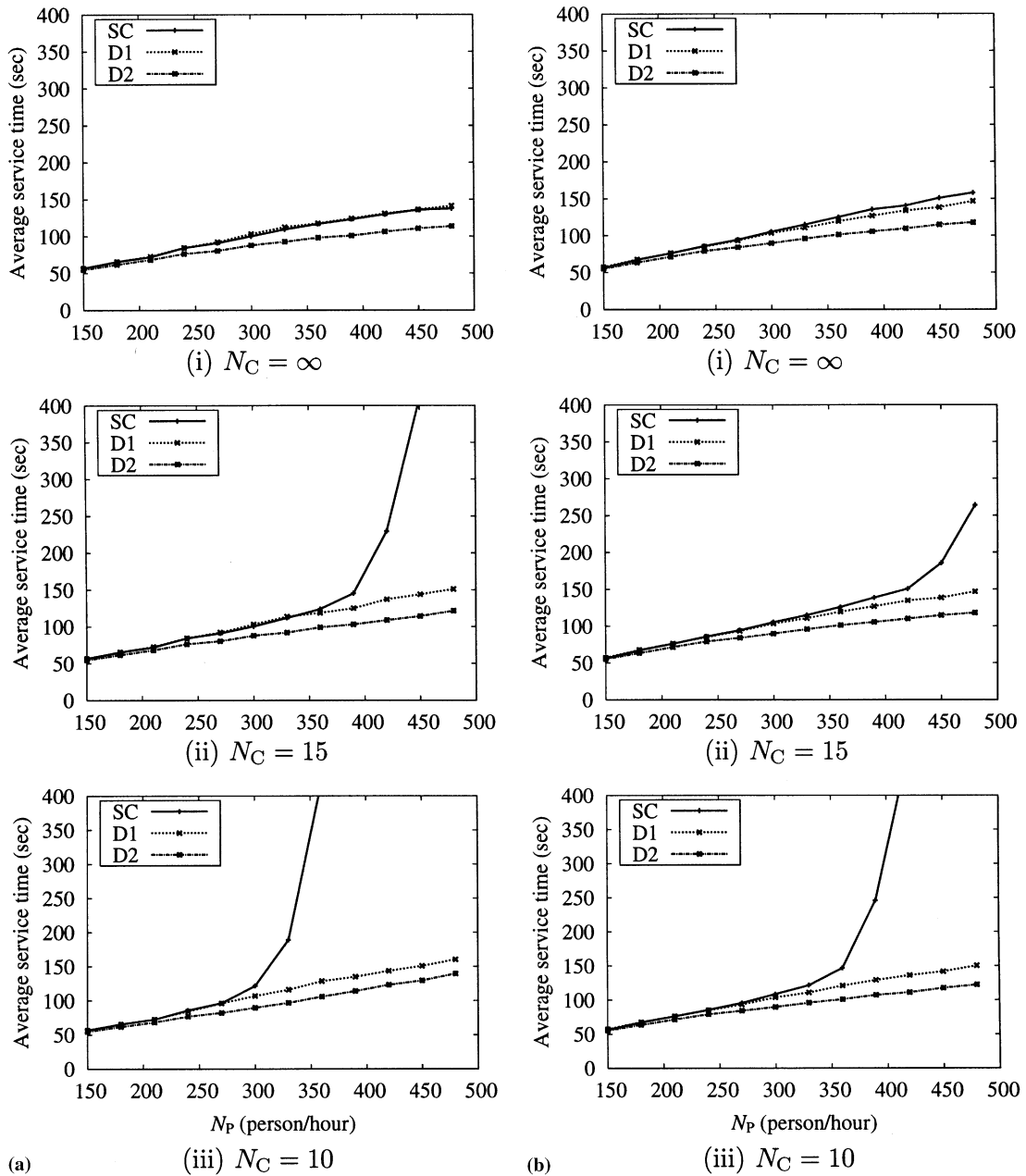


Fig. 3. Average service times (unweighted): (a) uppeak traffic and (b) downpeak traffic.

the destination floors of passengers who travel downwards are floor 1. It follows that passengers waiting at lower floors cannot board the car. If they cannot board the car in the next or later round trips, they are kept waiting and their service times deteriorate much. On the other hand, in dynamically optimized operation the car tries to serve passengers waiting at lower floors before those waiting at higher floors if the number

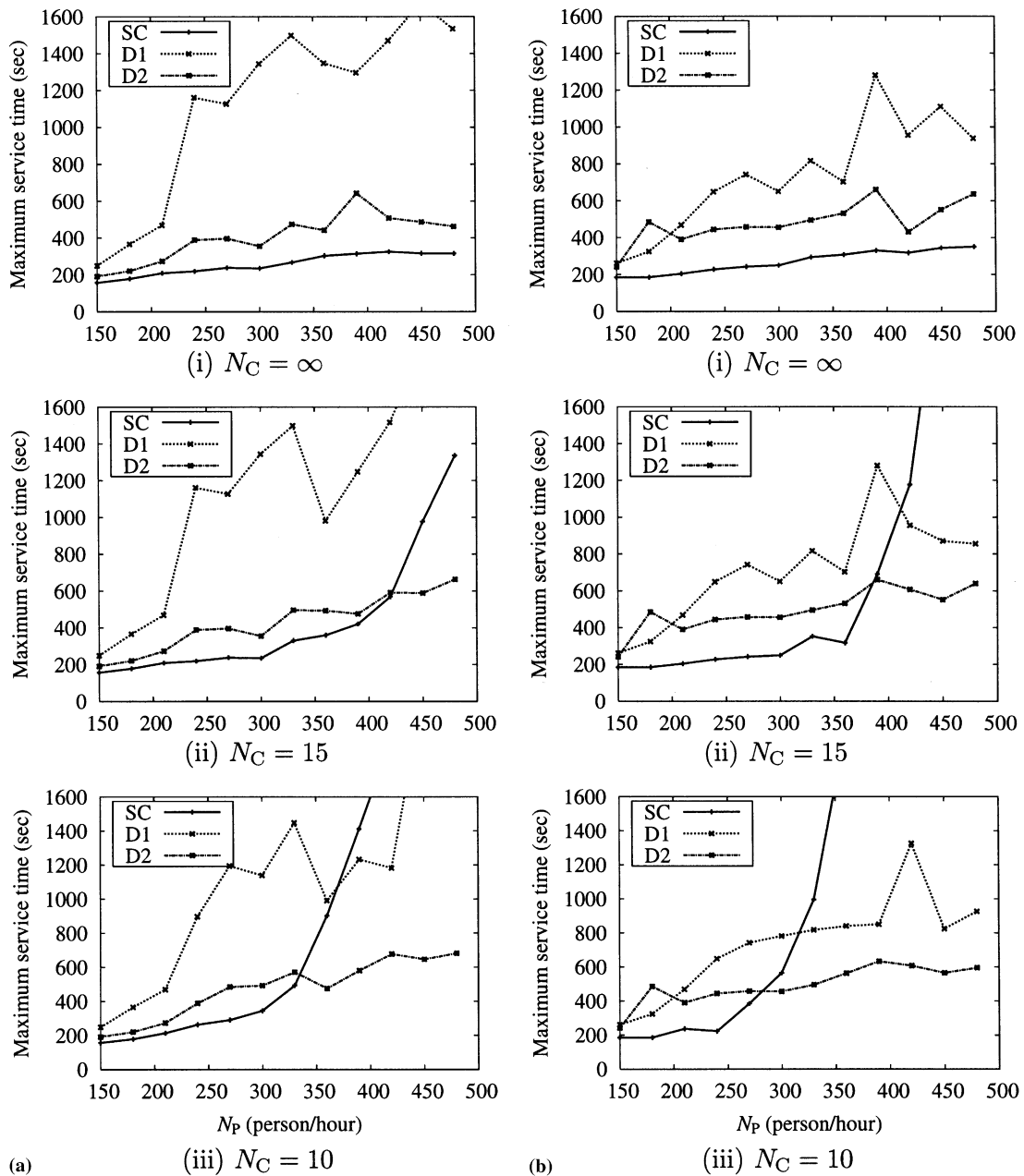


Fig. 4. Maximum service times (unweighted): (a) uppeak traffic and (b) downpeak traffic.

of passengers waiting at lower floors increases. Therefore, they have better transportation capability than selective collective operation.

Examples of car diagrams in SC and D1 during downpeak traffic are shown in Fig. 6, where $N_P = 960$ and $N_C = 10$. In the operation of Fig. 6(a), the car never stops at floor 2 while traveling downwards from 4000 to 5000 seconds. In this example, there are 10 passengers newly arrived at floor 2 during this period

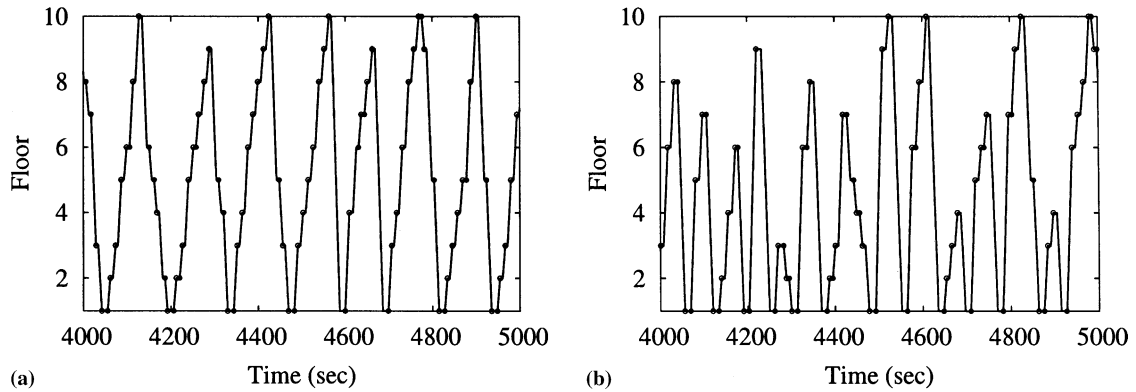


Fig. 5. Car diagrams in SC and D1 during uppeak traffic ($N_P=960$ and $N_C=10$): (a) selective collective operation (SC) and (b) dynamically optimized operation with reversal forbidden (D1).

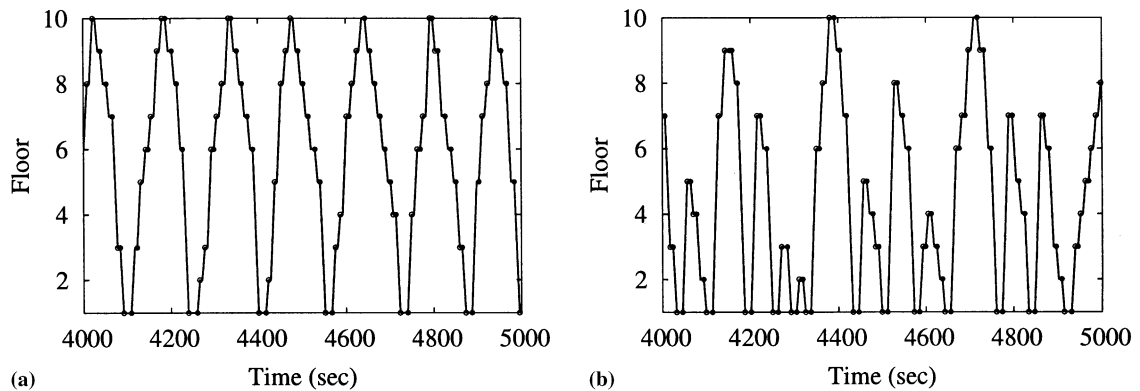


Fig. 6. Car diagrams in SC and D1 during downpeak traffic ($N_P=960$ and $N_C=10$): (a) selective collective operation (SC) and (b) dynamically optimized operation with reversal forbidden (D1).

whose destination floors are floor 1. Thus, all these 10 passengers are not served and kept waiting in the selective collective operation. This causes the deterioration of the average and maximum service times. On the other hand, in the dynamically optimized operation of Fig. 6(b), the car stops at floor 2 four times to load passengers while traveling downwards. This implies that the passengers waiting there are served well in this operation.

As for the maximum service times (Fig. 4), those in SC are smaller than those in D1 and D2 especially when N_P is small or N_C is large. As mentioned in Section 3.1, it is possible that the service for a particular passenger is deferred every time when operation is dynamically optimized if we choose the unweighted average service time as the objective functions of (P1) and (P2). This is confirmed by this simulation result.

4.2. Weighted case

As shown in the preceding subsection, if we choose the unweighted average service time as the objective functions of (P1) and (P2), the maximum service time tends to be larger in dynamically optimized opera-

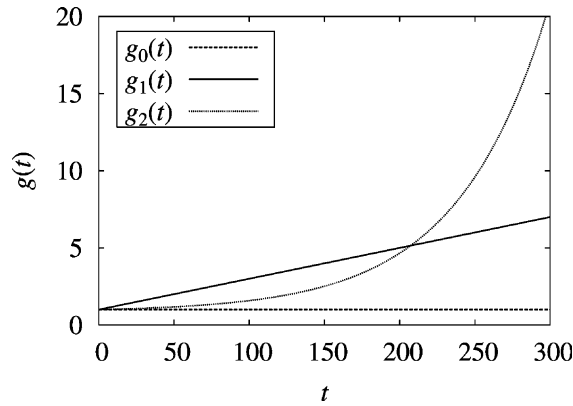


Fig. 7. Three types of weight functions.

tion. To avoid this, we choose the weighted average service time as the objective functions of (P1) and (P2). We adjust the weight μ_i for the passenger P_i according to his/her arrival time t_i and the optimization instant t_s ² as

$$\mu_i = g(t_s - t_i), \quad (28)$$

where $g(t)$ is an appropriate weight function. Here, we consider three types of weight functions:³

1. $g_0(t) = 1$. The unweighted case for comparison.
2. $g_1(t) = 1 + 0.02t$. A linear weight function.
3. $g_2(t) = 1 + \exp((t - 120)/60) - \exp(-2)$. An exponential weight function.

The shapes of these functions are shown in Fig. 7.

The simulation results are shown in Figs. 8–11. From these figures, we see that the maximum service time can be suppressed without deteriorating the average service time much by choosing appropriate weight functions. The exponential weight $g_2(t)$ is more effective than the linear weight $g_1(t)$ to suppress the maximum service time. With regard to the average service time, D1 with $g_2(t)$ is worse than that with $g_1(t)$ when passenger traffic is heavy, but D2 with $g_2(t)$ is rather better than that with $g_1(t)$. Therefore, it might be better to adopt the exponential weight $g_2(t)$ than to adopt the linear weight $g_1(t)$. We tested other linear weight functions with different slopes, but could not find a better linear weight function than $g_2(t)$.

It is quite interesting that when passenger traffic is light or the load capacity is large, SC yields better operation than D1 with $g_1(t)$ or $g_2(t)$, although D2 with $g_1(t)$ or $g_2(t)$ is better than or, at least as good as SC. More specifically, D2 is better than SC by about 10% with regard to the average service time, and is as good as SC with regard to the maximum service time, while D1 is as good as SC with regard to the average service time, but is worse than SC with regard to the maximum service time when passenger traffic is light or the load capacity is large. This indicates that dynamic optimization of the weighted average service time does not always result in the minimization of the average and maximum service times in overall operation when passenger traffic is so light that the car does not become full, although it is the most

² This t_s denotes the instant before applying the state reduction procedure described in Section 3.2.

³ Many other types of weight functions were tested, but only the results for typical functions are shown here.

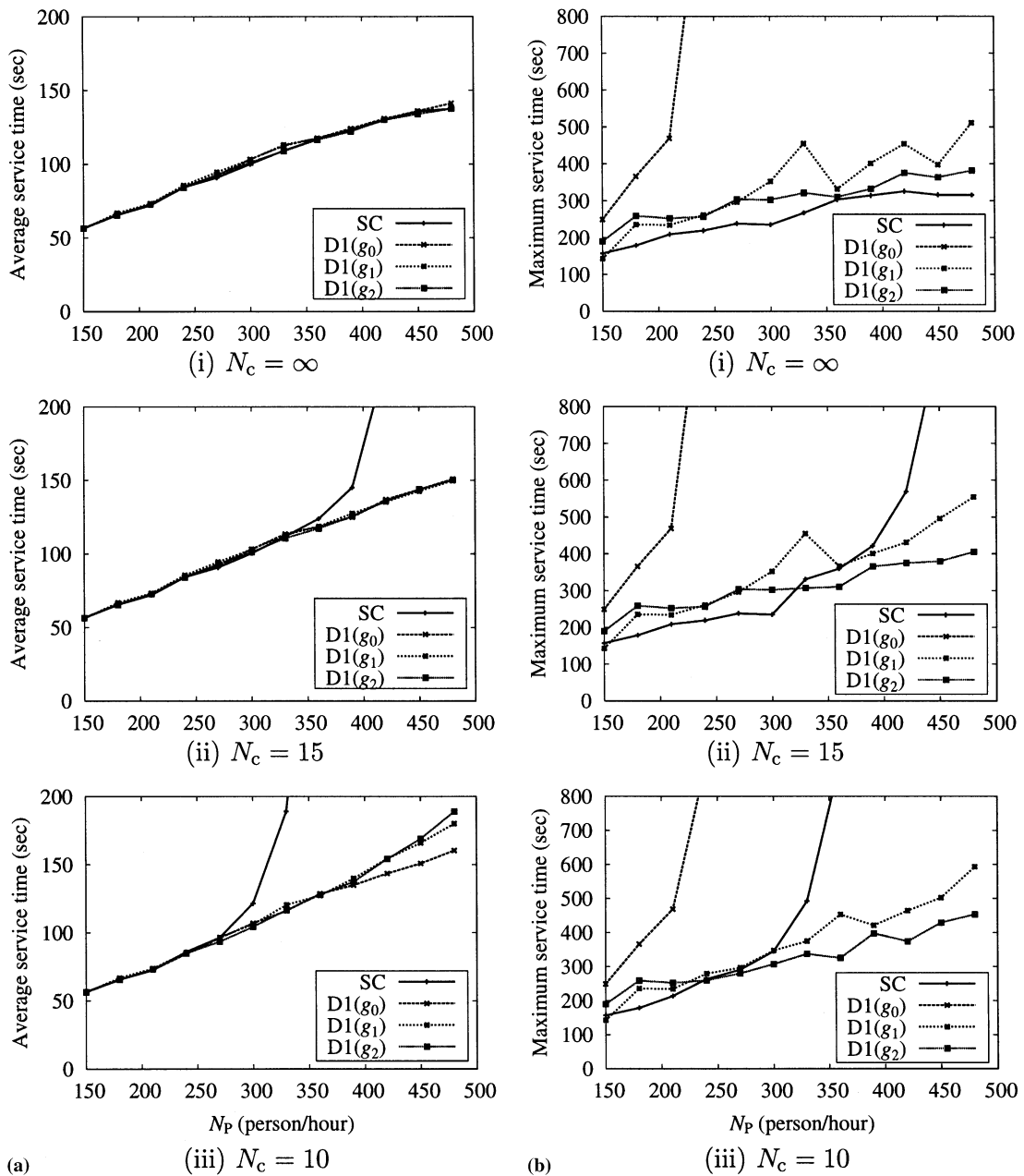


Fig. 8. Average and maximum service times (uppeak, weighted, reversal forbidden): (a) average service time and (b) maximum service time.

straightforward objective function for this purpose. This also suggests that if we are to design some operation heuristic for elevator systems with destination hall call registration, such a rule might be desirable that it works as the selective collective rule during light traffic and intimates dynamically optimized operation during heavy traffic.

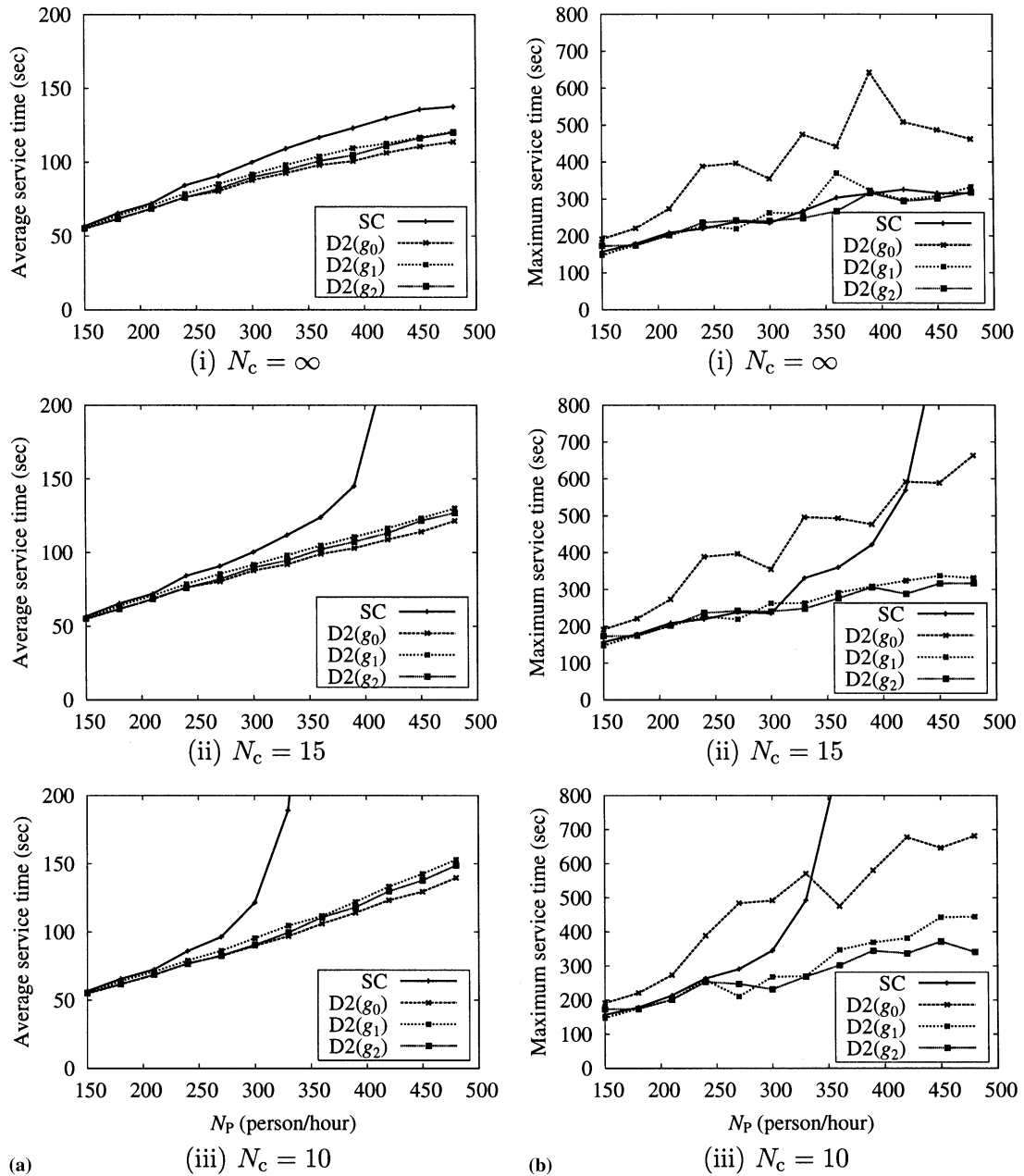


Fig. 9. Average and maximum service times (uppeak, weighted, reversal permitted): (a) average service time and (b) maximum service time.

The computational times and problem sizes (n) are shown in Table 3, where the weight function is chosen as $g_2(t)$. Although $2N_p$ optimization problems are to be solved in one simulation (for one combination of a data set and an operation type), only N_p problems solved within the interval [1800, 5400] are taken into consideration as in the case of the performance evaluation of operation. Thus, the maximum and average

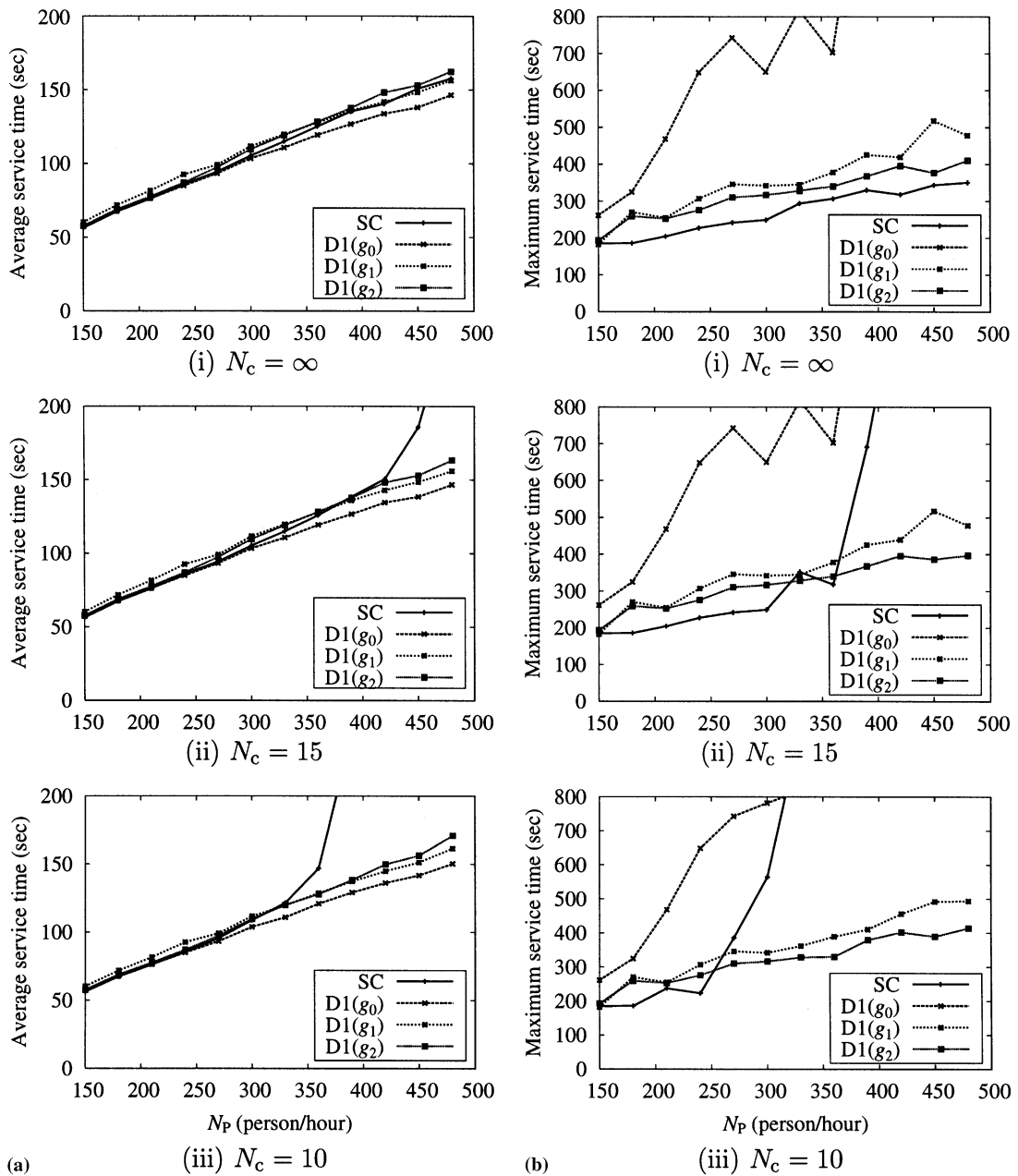


Fig. 10. Average and maximum service times (downpeak, weighted, reversal forbidden): (a) average service time and (b) maximum service time.

computational times or problem sizes shown in Table 3 are taken over 10 (data sets) $\times N_p$ (problems/data set) $= 10N_p$ (problems).

From this table, we can verify that our algorithm is fast enough for simulation-based operation analysis although it is not enough for real-time operation optimization. We can also see that the computational

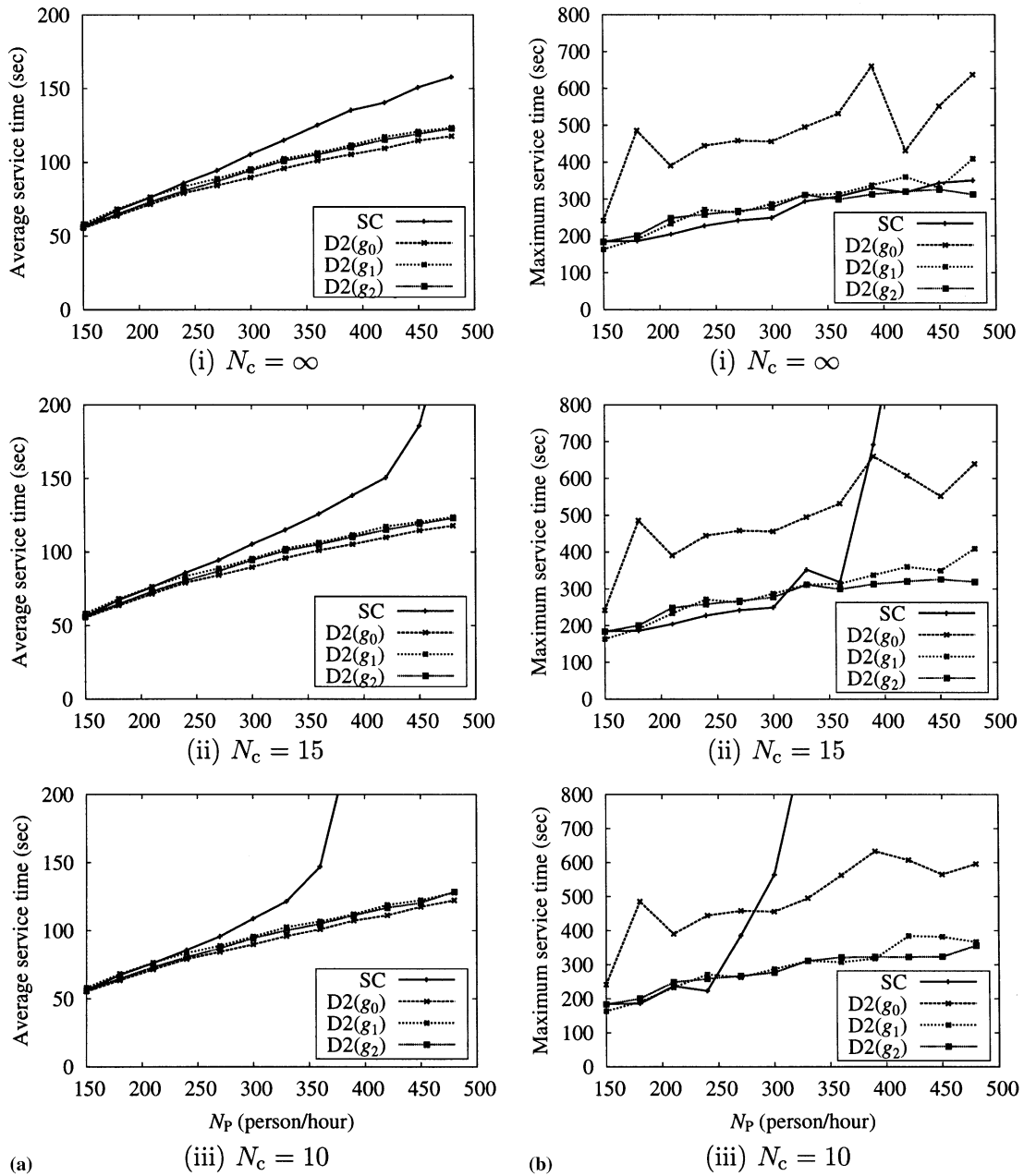


Fig. 11. Average and maximum service times (downpeak, weighted, reversal permitted): (a) average service time and (b) maximum service time.

times distribute widely depending on the sizes of the dynamically solved problems, i.e., the number of waiting and on-board passengers. If the problem sizes are the same, the branch-and-bound algorithm is in general faster for (P1) than for (P2) since the number of nodes generated for (P1) is smaller than that for (P2) because of the constraints forbidding reversal. However, D1 is less efficient than D2 and the sizes of the

Table 3
Computational times and problem sizes for $g_2(t)$

N_P	N_C	Traffic type	Computational time (second) ^a				Problem size ($=n$) ^b			
			D1		D2		D1		D2	
			Max.	Ave.	Max.	Ave.	Max.	Ave.	Max.	Ave.
420	∞	Uppeak	0.16	0.00	0.52	0.00	56	24.56	45	19.28
		Downpeak	0.48	0.01	0.28	0.00	60	30.92	53	22.60
	15	Uppeak	1.28	0.01	0.48	0.00	59	26.16	50	20.07
		Downpeak	0.47	0.01	0.28	0.00	60	30.93	55	22.54
	10	Uppeak	1.92	0.01	3.10	0.03	66	31.45	68	25.29
		Downpeak	1.19	0.01	0.70	0.01	60	31.41	48	22.96
450	∞	Uppeak	0.28	0.00	0.82	0.01	59	27.34	52	21.73
		Downpeak	1.62	0.01	1.30	0.01	71	34.09	52	24.90
	15	Uppeak	0.58	0.01	1.10	0.01	68	30.13	55	23.40
		Downpeak	1.62	0.01	1.30	0.01	71	34.03	52	24.90
	10	Uppeak	2.49	0.03	8.85	0.07	79	37.58	68	29.13
		Downpeak	1.25	0.01	0.88	0.01	69	34.96	53	25.26
480	∞	Uppeak	0.74	0.01	1.69	0.01	66	29.66	55	23.74
		Downpeak	6.28	0.02	2.67	0.01	74	38.40	52	27.11
	15	Uppeak	0.76	0.01	1.71	0.01	73	33.70	63	26.16
		Downpeak	6.28	0.02	2.68	0.01	71	38.60	52	27.22
	10	Uppeak	12.71	0.08	253.15	0.19	86	44.94	66	33.80
		Downpeak	1.14	0.02	4.72	0.02	75	40.77	55	28.76

^a The maximum and average computational times are taken over 10 (data sets) \times N_P (problems/data set) = $10N_P$ (problems).

^b The problem sizes are measured by n , the number of floors to be visited.

dynamically solved problems become larger in D1 than in D2. Therefore, as shown in Table 3, D1 requires longer computational times than D2 in many cases.

5. Conclusion

In this study we investigated the operation problem of single-car elevator systems with destination hall call registration. Our purpose here was to answer the two questions: “Can (single) car’s operation be improved with destination hall call registration?” and “If so, how can it be realized?” As an answer to the second question, we gave an explicit description of the single-car elevator operation problem and formulated it as a dynamic optimization problem such that the weighted average service time of passengers is minimized. Computer simulations revealed that dynamically optimized operation considerably improves the transportation capability compared to selective collective operation when passenger traffic is heavy. Now, we can answer “yes” to the first question.

Destination hall call registration with dynamic operation optimization has a potential to improve elevator performance even for single-car elevator systems. This implies that elevator systems with a small-size car can compete with those of a larger car if destination hall call registration is used. In the context of multi-car elevator systems, it follows that even the number of cars necessary to achieve desired performance can be reduced.

It would be necessary to find more appropriate objective functions for dynamic optimization since selective collective operation is yet better than dynamically optimized operation with reversal forbidden when

traffic is so light that the car does not become full, although dynamically optimized operation with reversal permitted is better than, or, at least as good as selective collective operation even in such cases. It might be a better way to change objective functions according to traffic situations.

Since strict dynamic optimization is not fast enough for real-time operation, some operation heuristic that replaces the selective collective rule is necessary for the application of our results to real elevator systems. As pointed out in Section 4, such an operation heuristic would be desirable that it works as the selective collective rule when passenger traffic is light, and imitates dynamically optimized operation when passenger traffic is heavy.

For our results to be applied to real elevator systems, it is also important to investigate the robustness of dynamically optimized operation with regard to passengers' unexpected behavior such as pushing wrong buttons, boarding the car without pushing buttons, and so on. When the car capacity is given by weights and the car cannot know exactly how many passengers it can load, robustness issues also arise because of the incompleteness of passenger information. To examine robustness for these uncertainties, it would be necessary to conduct computer simulations with more realistic situations taken into account. It might be easier to apply our results to vertical transportation systems for cargos or autonomous robots, which can be regarded as *honest*.

One of the most important topics left for future study is the extension of dynamically optimized operation to multi-car systems; to construct an algorithm to optimize passenger dispatching and each car's operation at the same time dynamically. If passenger dispatching should be re-optimized every time when a new hall call is registered, it is hard to apply our branch-and-bound algorithm directly to multi-car systems because each car's operation should be optimized for all possible combinations of cars and waiting passengers. It would be necessary to apply our algorithm in conjunction with, for example, the column generation approach. However, recent elevator control systems adopt such a dispatching policy that elevator systems inform passengers of which car they should board, immediately after they register their hall calls. It follows that once a car is dispatched to a passenger, it cannot be changed. Therefore, we only need to consider the dispatching of a newly arrived passenger when operation is re-optimized. Thus, it would be easy to extend our algorithm to multi-car systems with such a dispatching policy because the number of possible combinations of cars and passengers that should be considered is at most the number of cars.

Acknowledgements

The authors would like to show their best gratitude to Dr. S. Markon, Mr. M. Nakagawa and other members of Fujitec Co., Ltd. for their stimulating discussions and helpful advice.

Appendix A. Lemmas on Remark 3.3

Lemma A.1. *Let us denote by (P1a) and (P2a) the problems derived by the relaxation of (20) from (P1) and (P2), respectively. Then, all the optimal solutions of (P1a) and (P2a) satisfy (20).*

Proof of A.1. Assume that there is an optimal solution of (P1a) or (P2a) violating (20). It follows that in this solution there exist two passengers P_j and P_k such that $o_j = d_k$, and the i th and $(i+1)$ th in the visiting order of this solution are o_j (the origin floor of P_j) and d_k (the destination floor of P_k), respectively. Now, we construct another solution from this solution by exchanging the i th and the $(i+1)$ th in the visiting order. Then, the service time of P_k decreases by T_L while those of the other passengers are unchanged. It contradicts the assumption. \square

Lemma A.2. Let us denote by (P1b) and (P2b) the problems derived by the relaxation of (22) from (P1) and (P2), respectively. If μ_i ($P_i \in \mathcal{B} \cup \mathcal{W}$) satisfy

$$\mu_j \leq \mu_k \quad \text{if } t_j > t_k, \quad (\text{A.1})$$

there exist optimal solutions of (P1b) and (P2b) satisfying (22).

Proof of A.2. Assume that there is an optimal solution of (P1b) or (P2b) violating (22). It follows that in this solution there exist two passengers P_j and P_k such that $d_j = d_k$ and $t_j > t_k$, and that the i th and $(i+1)$ th in the visiting order of this solution are d_j (the destination floor of P_j) and d_k (the destination floor of P_k), respectively. Now, we construct another solution from this solution by exchanging the i th and the $(i+1)$ th in the visiting order. Then, the service times of P_j and P_k increase by T_L and $-T_L$, respectively, while those of the other passengers are unchanged. Therefore, the total increase of the objective function value is given by $(\mu_j - \mu_k)T_L$. Since $t_j > t_k$ and thus $\mu_j \leq \mu_k$ from the assumption of the lemma, this solution should also be optimal ($\mu_j = \mu_k$). Thus, we can construct an optimal solution satisfying (22) by applying this exchange repeatedly. \square

Appendix B. Selective collective operation

In this section we explain selective collective operation briefly. Some types of multi-car elevator operation such as duplex collective operation [10] is often referred to as “selective collective operation.” However, in this study we call the most commonly used single-car elevator operation as selective collective operation. Even in multi-car systems, each car’s operation is determined by (single-car) selective collective operation after passengers are dispatched to cars.

In selective collective operation, the car serves passengers via the following simple rule.

1. If the car is empty and stays at a floor when a new hall call is registered, it starts moving toward the floor where the passenger is waiting.
2. The car serves passengers who are waiting ahead of the car and whose directions are equal to the travel direction of the car. The travel direction of the car should not be changed until the car finishes serving all such passengers.

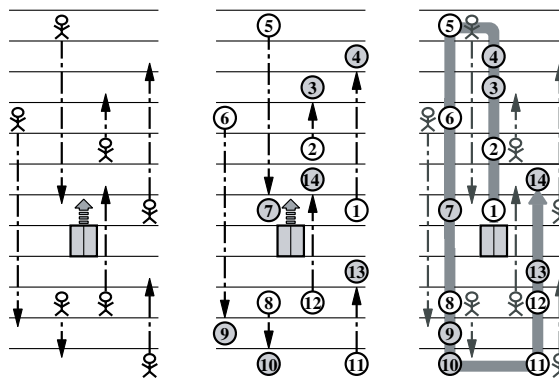


Fig. 12. An example of selective collective operation.

3. When the car finishes serving all such passengers, the car starts serving passengers in the opposite direction. For example, if the current travel direction of the car is upward, the car travels to the highest floor where passengers who are going downward are waiting, reverses its direction, and starts serving them.

Repeat this until no passengers are left unserved.

An example of selective collective operation is shown in Fig. 12.

In selective collective operation, only the directions of waiting passengers are necessary and reversal never occurs.

References

- [1] W.L. Chan, A.T.P. So, Dynamic zoning for intelligent supervisory control, *International Journal of Elevator Engineering* 1 (1996) 47–59.
- [2] R.H. Crites, A.G. Barto, Elevator group control using multiple reinforcement learning agents, *Machine Learning* 33 (1998) 235–262.
- [3] S. Hikita, K. Komaya, A new elevator group-supervisory control system using fuzzy rule-base (in Japanese), *Transactions of the Society of Instrument and Control Engineers* 25 (1) (1989) 99–104.
- [4] H. Lee-Kwang, C.B. Kim, Techniques and applications of fuzzy theory to an elevator group control system, in: C.T. Leondes (Ed.), *Fuzzy Theory and Systems: Techniques and Applications*, vol. 1, Academic Press, New York, 1999, pp. 461–481.
- [5] D. Levy, M. Yadin, A. Alexandrovits, Optimal control of elevators, *International Journal of Systems Science* 8 (3) (1977) 310–320.
- [6] S. Markon, H. Kita, Y. Nishikawa, Adaptive optimal elevator group control by use of neural networks, *Transactions of the Institute of Systems, Control and Information Engineers* 7 (12) (1994) 487–497.
- [7] D.L. Pepyne, C.G. Cassandras, Optimal dispatching control for elevator systems during uppeak traffic, *IEEE Transactions on Control Systems Technology* 5 (6) (1997) 629–643.
- [8] D.L. Pepyne, C.G. Cassandras, Design and implementation of an adaptive dispatching controller for elevator systems during uppeak traffic, *IEEE Transactions on Control Systems Technology* 6 (5) (1998) 635–650.
- [9] T. Inamoto, H. Tamaki, H. Murao, S. Kitamura, Deterministic optimization model of elevator operation problems and an application of branch-and-bound method (in Japanese), *The Transactions of the Institute of Electrical Engineers of Japan* 123-C (7) (2003) 1334–1340.
- [10] J.B. Halpern, L.E. Pike, Elevator operation and control, in: G.R. Strakosch (Ed.), 3rd ed., *Vertical Transportation Handbook*, John Wiley and Sons, New York, 1998, pp. 129–167.
- [11] H.N. Psaraftis, A dynamic programming solution to the single vehicle many-to-many immediate request dial-a-ride problem, *Transportation Science* 14 (2) (1980) 130–154.