COLLISION AVOIDANCE METHOD FOR MULTI-CAR ELEVATOR SYSTEMS WITH MORE THAN TWO CARS IN EACH SHAFT

Shunji Tanaka[a], Daiki Hoshino[b]

[a,b]Department of Electrical Engineering, Kyoto University

[a]tanaka@kuee.kyoto-u.ac.jp, [b]hoshino@rotary.kuee.kyoto-u.ac.jp

ABSTRACT
In multi-car elevator systems, several cars are installed in each elevator shaft to improve the transportation capability without increasing the occupied floor space. In our previous studies, we proposed an optimization-based method to avoid collisions between cars in the same shaft of multi-car elevator systems. However, it was applicable only to the systems with two cars in each shaft. In this study we will improve this method so that it can handle more than two cars. Then, its effectiveness will be examined by computer simulation.

Keywords: multi-car elevator system, collision avoidance, optimization-based method, computer simulation

1. INTRODUCTION

A multi-car elevator system is such an elevator system that several cars are installed in every elevator shaft. This type of elevator system has been attracting much attention (ThyssenKrupp Elevator 2005, Miyamoto and Yamaguchi 2008, Onat et al. 2011, Valdivielso and Miyamoto 2011) because it enables us to improve the vertical transportation capability with less floor space compared to the ordinary elevator systems. However, intelligent car group control is crucial to take full advantage of multi-car elevator systems.

The roles of a group control system are (1) to allocate calls (passengers) to cars and (2) to control cars so that collisions never occur. Miyamoto and Yamaguchi (2008) constructed a simulator of a multi-car elevator system called MceSim for CST Solution Competition held in Japan. Based on this simulator, Valdivielso and Miyamoto (2011) proposed call allocation methods. Although MceSim has a function to avoid collisions, this simulator is based on a simplified model of a multi-car elevator system because their focus was rather on call allocation methods. For example, the car stop time at a floor is constant regardless of the number of passengers who board or leave the car, a car can change its speed only step-wise every one second, and so on. On the other hand, our previous studies considered a more realistic multi-car elevator system where the car stop time cannot be known in advance by the system, and the acceleration and the jerk of cars are simulated exactly. In Tanaka and Watanabe (2009, 2010), we proposed a collision avoidance algorithm for this realistic system that dynamically optimizes the floors to be visited next by cars. Then, in Tanaka and Miyoshi (2011), two types of passenger guidance methods, immediate guidance and nonimmediate guidance were compared by computer simulation. However, the collision avoidance algorithm in our previous studies was not applicable to a system with more than two cars in each shaft.

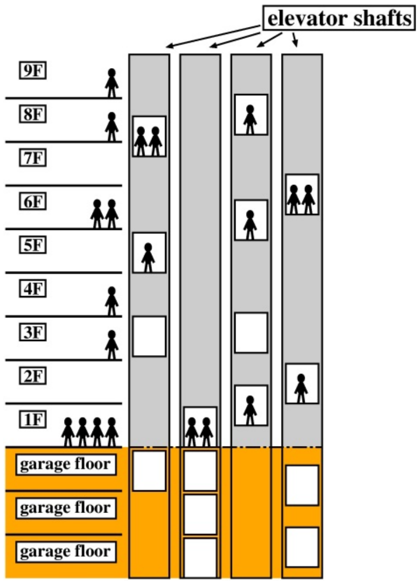
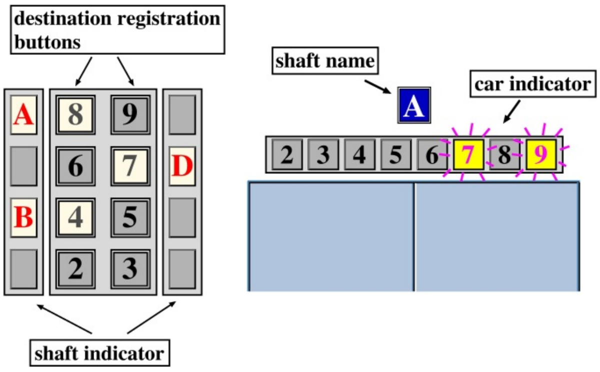


Figure 1: Multi-Car Elevator System

Our purpose in this study is to extend this algorithm so that it can handle more than two cars. Then, its effectiveness will be verified by computer simulation.

2. SYSTEM CONFIGURATION

Figure 1 gives a typical configuration of a multi-car elevator system. The first floor is the ground floor and there are garage floors underground so that the lower cars can escape to there when the upper cars are going to stop at the ground floor. On the other hand, there are no garage floors assumed on the top of the building. It follows that only the uppermost car in each shaft can stop at the highest floor. Therefore, the system should determine the assignment of cars to each passenger



(a) Destination registration buttons (b) Destination-based car guidance

Figure 2: Interface for Passengers

according to his/her destination floor. To achieve this, a destination-based call registration system is installed in multi-car elevator systems. A passenger who comes to an elevator hall calls a car by pushing the button of his/her destination floor (Fig. 2(a)) if it has not been pushed by another passenger. Then, the system immediately displays the shaft where he/she should wait for a car on the shaft guidance panel. In other words, we adopt the immediate passenger guidance policy. Finally, the car arrives at the floor and the car guidance panel displays the destination floors of passengers who should board that car (Fig. 2(b)).

The group control system for this multi-car elevator system is composed of a group controller and shaft controllers as shown in Fig. 3. When a new call is registered, the group controller immediately allocates it to some car and generates a travel schedule of that car by the selective-collective rule (Strakosch 1998), i.e. the ordinary elevator car operation, without considering collisions. This travel schedule is given by a visiting order of the origin and the destination floors of the calls allocated to that car. The travel schedules are passed to shaft controllers and they are modified so that collisions and reversal do not occur. Here, reversal is such an undesirable operation that a car travels in the direction opposite to the desired direction of on-board passengers. The shaft controllers should keep the visiting orders of the origin and destination floors in the travel schedules by the group controller, but it is possible that reversal cannot be avoided without modifying them. Hence, the visiting orders are modified only in such a case. It follows that the collision avoidance is performed mainly by inserting into the travel schedules, waits and/or evacuations to the floors where no passengers board nor leave the cars.

Since we do not assume that all the passengers push buttons to register their destination floors, the system cannot know how many passengers correspond to one call. It implies that the system cannot know the car stop time at a floor because how long it takes for a car to load or unload passengers at that floor cannot be known in advance. This setting is reasonable even when such a system is adopted that all the passengers

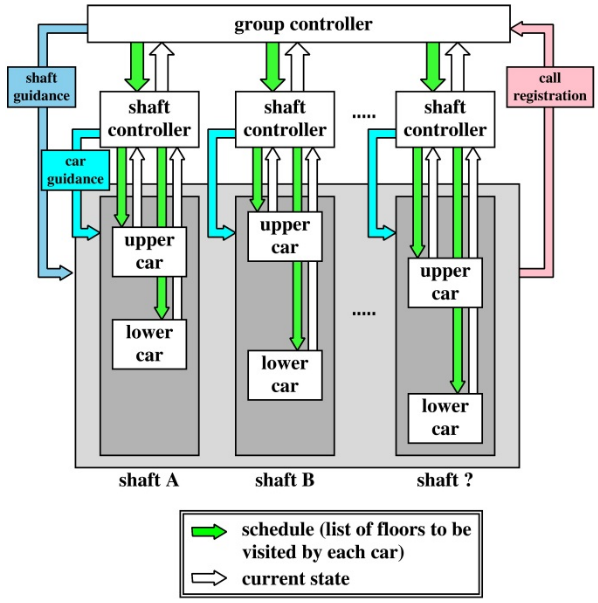


Figure 3: Group Control System for Multi-Car Elevator

push buttons and register their destination floors, because passenger boarding and leaving times vary in practice. Under this realistic setting, we are to consider a collision (and reversal) avoidance method.

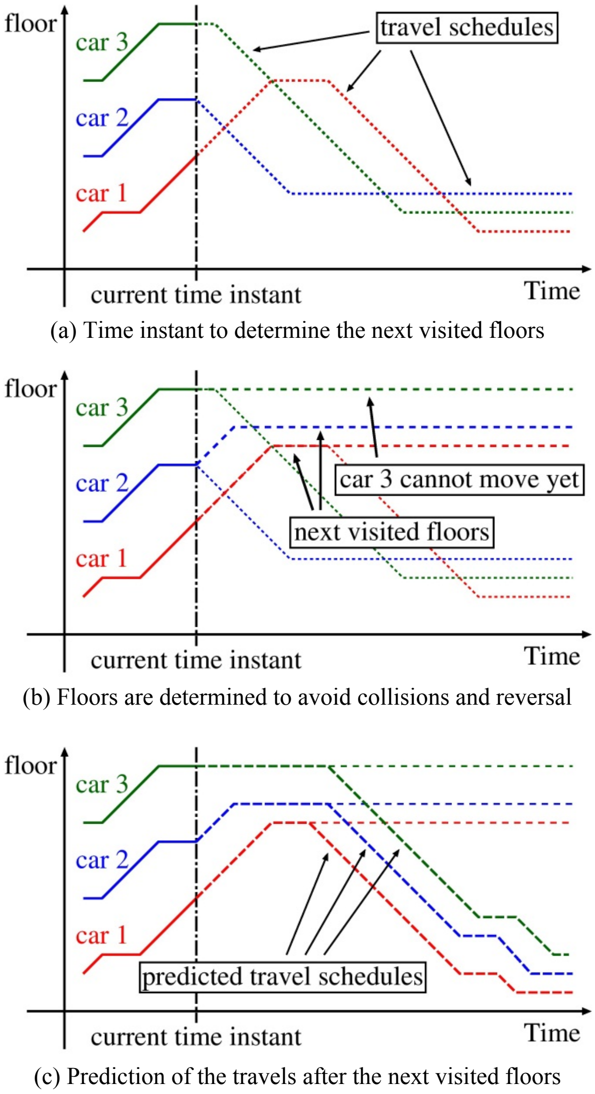
3. OPTIMIZATION-BASED COLLISION AVOIDANCE

The framework of the collision avoidance method in this study is the same as that in our previous studies (Tanaka and Watanabe 2009, 2010). In this method, collisions are considered only until cars reach the floors to be visited next, and the cars are assumed to stay there infinitely long. The collision avoidance is considered again when one of the cars becomes ready to start. This is repeated until all the passengers are served.

The collision avoidance algorithm is triggered at the following instants:
(a) A new call is allocated to one of the cars in the shaft.
(b) A moving car arrives and stops at a floor.
(c) A car finishes closing the door and becomes ready to start.

Figures 4(a) and 4(b) depict an example of our collision avoidance method. Since collisions occur in the travel schedules passed from the group controller, they are modified at the instant when the collision avoidance algorithm is triggered. At this instant, car 3 cannot move yet and hence is assumed to stay at the current floor infinitely long, while car 1 and car 2, which are already moving or can start moving, are assumed to stay at the next visited floors infinitely long.

To achieve "good" collision avoidance, all the feasible combinations of the next visited floors such that collisions and reversal never occur are enumerated and the best combination that minimizes several types of objective functions is adopted. To evaluate one of the objective functions, future travel schedules after arriving at the next visited floors are necessary. To predict them as precisely as possible, the travel



(a) Time instant to determine the next visited floors

(b) Floors are determined to avoid collisions and reversal

(c) Prediction of the travels after the next visited floors

Figure 4: Concept of Dynamic Optimization for Collision Avoidance

schedules given by the group controller are modified by a collision avoidance rule and the obtained schedules are used (Fig. 4(c)). The primary reason why the method in our previous studies was not applicable to a system with more than two cars in each shaft is that this rule can consider only two cars. Therefore, in the next section, we will propose a new collision avoidance rule for the travel schedule prediction that is applicable to any number of cars.

4. COLLISION AVOIDANCE RULE FOR PREDICTION OF FUTURE CAR TRAVEL

The collision avoidance rule in our previous studies determines which car should wait and/or be evacuated based on the detailed model of car motion, i.e. by taking into account the acceleration and the jerk of a car. However, it is difficult to extend this to the case with more than two cars because the collision avoidance becomes much more complicated. Therefore, in this study, we propose a two-phase method as shown in Fig. 5. First, collision avoidance and reversal avoidance are

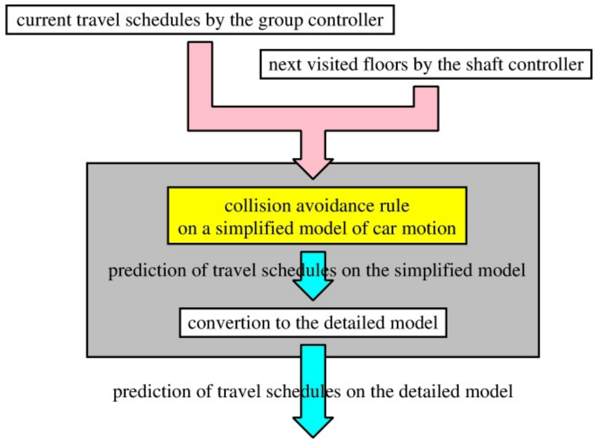


Figure 5: Prediction of Future Car Travel by a Collision Avoidance Rule

considered on a simplified model of car motion and then, the obtained travel schedules are converted by inserting appropriate waits so that collisions and reversal never occur on the detailed model. On the simplified model, the following assumptions are made:

(1) A car moves one floor in a unit time.
(2) A car can stop anytime.
(3) The floor stop time is zero.
(4) A collision occurs when more than one car occupies the same floor at a time.

Obviously, these assumptions make the collision avoidance much easier because the difference between the current position (floor) of a car and its previous position should be +1, 0 or -1 and hence we can move it step by step.

The outline of the collision avoidance rule for n_{C} cars (car 1, car 2, …, car n_{C} from the lowest to the highest) on the simplified model is as follows.

(0) Let the current time instant t be $t := 0$ and denote the initial position of car i by s_{i0}. Let f_i^{turn} ($i = 0, \ldots, n_{\mathrm{C}}$) be the floor where car i changes its direction for the first time in the travel schedule. If the travel schedule finishes without any direction change, let f_i^{turn} be the final floor in the travel schedule. If the travel schedule is empty, let $f_i^{\mathrm{turn}} := \emptyset$. Let d_i denote the current destination of car i. If car i is not empty, it is initialized by the floor where it becomes empty for the first time in the travel schedule. Otherwise, $d_i := \emptyset$.

(1) For each car i such that $d_i \neq f_i^{\mathrm{turn}}$, check if the car can reach f_i^{turn} without changing its direction. If there is more than one candidate, such a car is chosen first that interferes the less number of cars with $d_i \neq f_i^{\mathrm{turn}}$.

(2) Determine the next positions of cars at $t+1$, $s_{i,t+1}$, for the cars i with $d_i \neq \emptyset$ so that they approach their destinations d_i unless collisions nor reversal occur. In the case that a collision or reversal occurs between a pair of cars, the car nearest to its destination d_i is moved and the other car is kept waiting at the current position s_{it}.

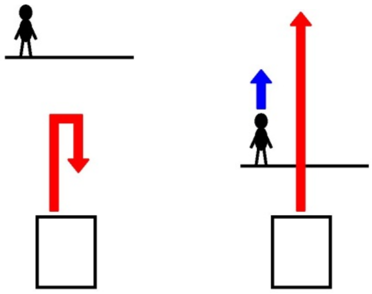


Figure 6: Unnatural Car Operations

(3) The next positions $s_{i,t+1}$ of cars i with $d_i = \emptyset$ are chosen from among $s_{it} - 1, s_{it}$ and $s_{it} + 1$ so that collisions never occur.
(4) Let $t \coloneqq t + 1$. For the cars i with $s_{it} = d_i$, let $d_i \coloneqq \emptyset$. Update f_i^{turn} if $s_{it} = f_i^{\text{turn}}$.
(5) If $f_i^{\text{turn}} = \emptyset$ for all i, terminate. Otherwise, go to (1).

Roughly speaking, this rule moves each car to the floor where it changes its direction next in the travel schedule while ensuring that collisions and reversal never occur. The following condition checks whether reversal becomes unavoidable between a pair of cars i and j ($i < j$):

$$s_{it} + j - i > d_j \land d_i + j - i > s_{jt}. \quad (1)$$

It is similar to the condition in Tanaka and Watanabe 2010 except the term $j - i$ to take into account the cars between the two cars.

The original travel schedules generated by the group controller are modified by the above rule so that collisions and reversal do not occur on the simplified model. The obtained travel schedule of a car is given by a visiting order of floors where some additional floors for evacuation are inserted into the original one. To convert such travel schedules into those on the detailed model, waits at the visited floors are inserted without changing the visiting orders of the floors so that collisions do not occur.

5. OBJECTIVE FUNCTIONS

As explained in Section 3, the next visited floors are evaluated by several types of objective functions in our collision avoidance method. These objective functions will be briefly introduced in this section.

In the proposed method, the following five types of objective functions are employed.

(1) The number of such cars that the visiting orders of the floors in the travel schedules are modified to avoid reversal. This objective function is to reduce unnecessary modifications of the visiting orders as much as possible (see Section 2).
(2) The number of empty cars that satisfy one of the following conditions.
   (a) It changes the direction although it is approaching the floor scheduled next in the travel schedule.
   (b) It skips the floor to be visited next in the travel schedule and its direction is the same as that of the passengers who are waiting for the car at that floor.

   This objective function is to suppress unnatural operations (see Fig. 6).
(3) Total service time of passengers. The service time of a passenger is the time from when he/she comes to the elevator hall of his/her origin floor until when he/she finishes leaving a car at his/her destination floor. Since the number of passengers that correspond to one call is unknown to the system, it is assumed to be one. This objective function is to achieve as good collision avoidance as possible.
(4) Total absolute difference between the next visited floor and the floor to be visited next in the travel schedule. This objective function is to follow the travel schedules by the group controller as much as possible.
(5) Total absolute difference between the current floor and the next visited floor for the cars without allocated calls. This objective function is to suppress unnecessary travels of empty cars.

These objective functions are evaluated in the lexicographical order. More specifically, if (1) is the same, (2) is evaluated and if (2) is the same, (3) is evaluated, and so on. Among them, (3) requires the future travel schedules after the next visited floors.

6. COMPUTER SIMULATION

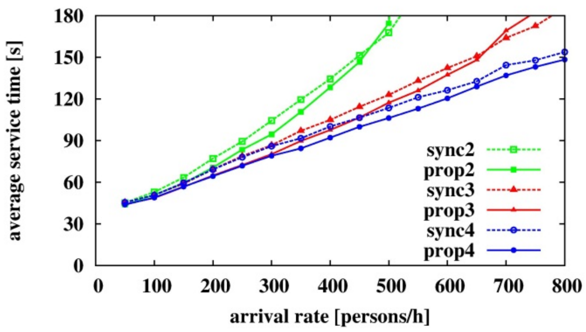
In this section the effectiveness of the proposed collision avoidance method will be examined by computer simulation. The specifications of the system follow Tanaka and Watanabe (2009, 2010), which are summarized in Table 1.

Table 1: System Specifications

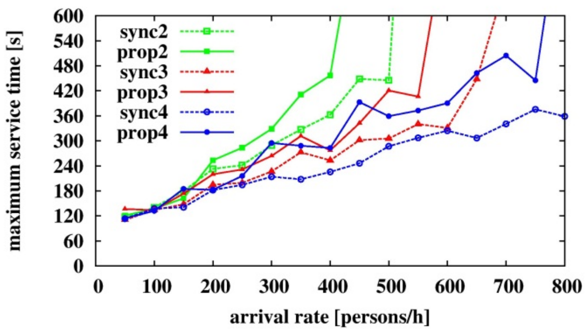
Number of floors	30
Interfloor distance	4.33m
Passenger boarding/leaving time	1.2s/person
Passenger response time	2.0s
Door opening time	1.8s
Door closing time	2.4s
Maximum speed	6m/s
Maximum acceleration	1.1m/s^2
Jerk	2.0m/s^3
Car capacity	20persons

Data sets of passengers are generated as follows. Passenger arrival times at elevator halls are generated from the uniform distribution [0, 7200) (in seconds). Their origin and destination floors are randomly generated to simulate two types of passenger traffic, uppeak traffic and downpeak traffic, and the ratios among

(1) the number of passengers from the 1st floor,
(2) the number of passengers to the 1st floor,
(3) the number of upward passengers that travel between floors other than the 1st floor,

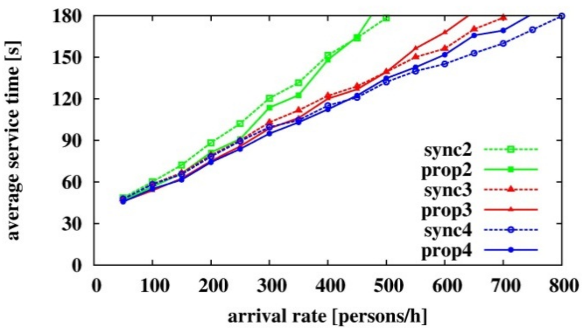


(a) Average service time

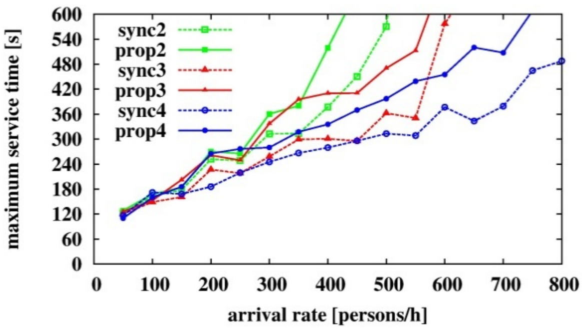


(b) Maximum service time

Figure 7: Results for Single-shaft System (Uppeak)



(a) Average service time



(b) Maximum service time

Figure 8: Results for Single-shaft System (Downpeak)

(4) the number of downward passengers that travel between floors other than the 1st floor,

are set to (1):(2):(3):(4)=30:1:1:1 and 1:30:1:1, respectively. For each setting of the passenger arrival rate (the number of passengers per one hour) and the type of passenger traffic, 10 data sets are generated. To examine stationary performance, the average and maximum service times of passengers whose arrival times are in the interval [1800, 5400) are evaluated.

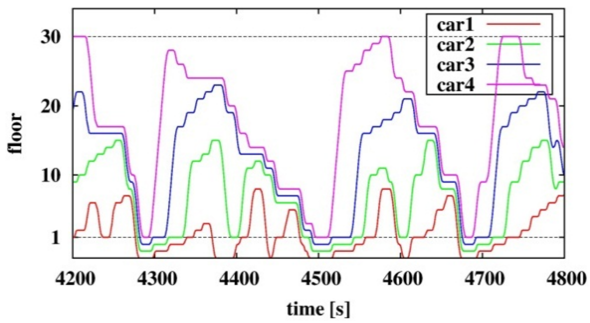
First, the simulation results for a system with a single shaft are shown. For comparison, we considered synchronous control (Valdivielso and Miyamoto 2011). This control restricts the directions of the cars in the same shaft so that they are always the same to make the collision avoidance easy. The allocation of a call to a car is determined by zoning (Strakosch 1998). It determines which car should serve a call by its origin floor (O_k) and destination floor (D_k) as follows. Let us denote the number of cars by n_C and number the cars from the lower to the upper as car 1, …, car n_C. Let us also denote the highest floor of the building by the M-th floor. If the direction of the call k is up, i.e. $O_k < D_k$, the call is allocated to car $\lceil n_\mathrm{C} D_k / M \rceil$, where $\lceil a \rceil$ denotes the smallest integer that is not smaller than a. On the other hand, if the direction of the call is down, the call is allocated to car $\lceil n_\mathrm{C} O_k / M \rceil$. In other words, the floors are equally divided into zones and each car serves only calls in its zone.

The results are summarized in Figs. 7 and 8. In these figures, the average or the maximum service time over 10 instances is depicted against the passenger arrival rate. In addition, "prop" and "sync" denote the results of the proposed method and the synchronous control, respectively, and the numbers at the tail of "prop" and "sync" denote the numbers of cars. From these figures, we can verify that the proposed method can improve the average service time compared to the synchronous control. However, the maximum service time is worse for both the types of traffic.

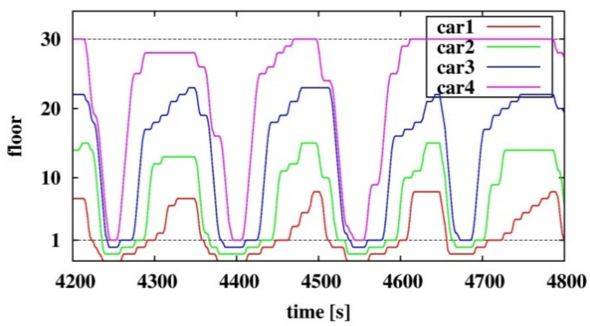
Figures 9(a) and 9(b) are examples of the obtained car diagrams by the proposed method and the synchronous control, respectively. From these figures, we can observe the reason why the maximum service time diverges for a smaller passenger arrival rate when the proposed method is applied. It is because the round trip time, i.e. the cycle of returning to the 1st floor becomes larger for upper cars compared to the synchronous control. Due to this larger round trip time, the number of passengers waiting for an upper car at the 1st floor reaches the car capacity in uppeak traffic. In this case, the car becomes full at the 1st floor and cannot serve other calls after that until at least one passenger leaves the car. Therefore, upward calls with lower origin floors are likely to be delayed and the maximum service time diverges. For downpeak traffic, a similar argument holds.

Next, computer simulation for a system with five shafts is performed. Since there is more than one shaft, we should allocate each call to some shaft, while car allocation in the shaft is determined by zoning. The shaft allocation is determined by the following simple algorithm (Tanaka and Watanabe 2009, 2010).

(1) For each shaft 1, 2, …, 5:



(a) Proposed method



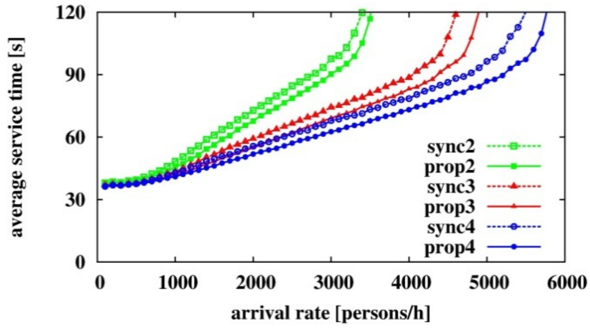
(b) Synchronous control

Figure 9: Examples of Car Diagrams (Four Cars, Uppeak)

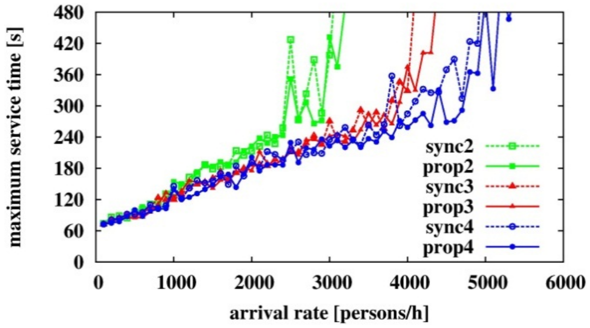
(i) Apply the collision avoidance rule in Section 4 to the current travel schedules of the cars in the shaft and compute the total service time in the shaft.
(ii) Allocate the call to the car in the shaft by zoning, and obtain the travel schedule by the selective-collective rule.
(iii) Apply the collision avoidance rule to the new travel schedules and compute the total service time.
(iv) Compute the increase of the total service time from (i) to (iii).

(2) Choose the shaft with the least increase of the total service time.

The results are shown in Figs. 10 and 11. The improvement from the synchronous control is more apparent in this case. This implies that the proposed method performs very well when the number of shafts increases and several cars in different shafts share one zone. To see this, the performance of the two methods is compared with the number of shafts changed. The results for $n_C = 4$ and downpeak traffic are shown in Fig. 12. In these figures, the horizontal axes are normalized by the number of shafts. Therefore, the unit is given by persons/h/shaft. In addition, the numbers before "sync" and "prop" denote the numbers of shafts. From this figure, we can see that the transportation capability per one shaft improves as the number of shafts increases and the improvement is larger for the proposed method than for the synchronized control. This would be because cars in different shafts are more likely to play different roles in the proposed method than those in the synchronous control that restricts the direction in each shaft and hence has less freedom of operation. This advantage is thought to contribute to the improvement of the transportation capability.

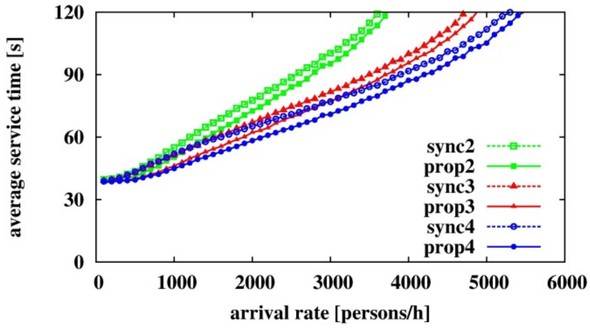


(a) Average service time

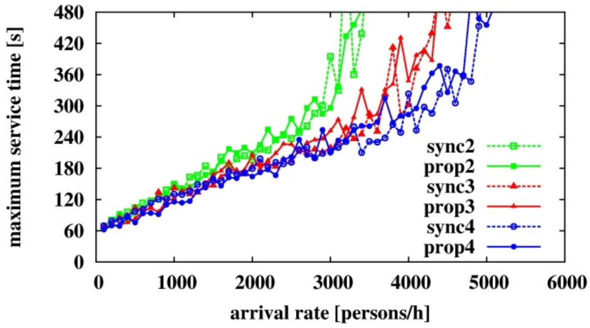


(b) Maximum service time

Figure 10: Results for Five-shaft System (Uppeak)



(a) Average service time

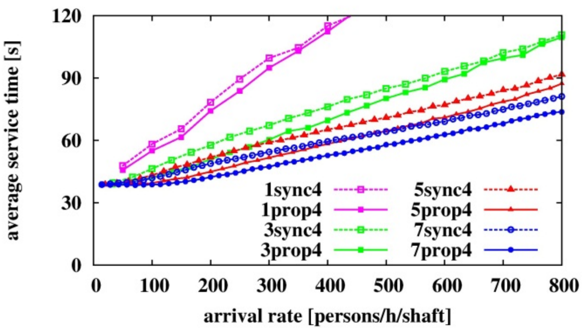


(b) Maximum service time

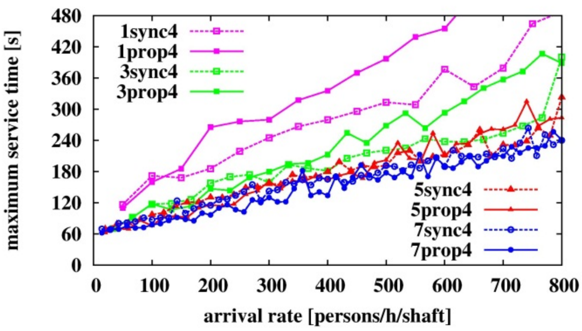
Figure 11: Results for Five-shaft System (Downpeak)

7. CONCLUSION

In this study we extended the previous collision avoidance method for multi-car elevator systems so that



(a) Average service time



(b) Maximum service time

Figure 12: Results When the Number of Shafts Are Changed (Number of Cars $n_C = 4$, Downpeak)

it becomes applicable to the systems with more than two cars in each shaft. Then, computer simulation was conducted to examine the effectiveness of the proposed method through the comparison with the synchronous control. As a result, it was verified that the proposed method can improve the transportation capability especially when the number of shafts is more than one, compared to the synchronous control. However, the maximum service time tends to increase because the proposed method does not try to minimize it. To improve the method for suppressing the maximum service time is left for future research. It would also be necessary to investigate more intelligent car allocation algorithms.

ACKNOWLEDGMENTS

This work is partially supported by Grant-in-Aid for Scientific Research (C) 23560483, from the Japan Society for the Promotion of Science (JSPS).

REFERENCES

ThyssenKrupp Elevator. 2005. *Two elevator cabs have always meant two shafts. Until now. TWIN -- A singular revolution in elevator design.* Available from: http://www.thyssenkruppelevator.com/downloads/TWIN.pdf [July, 2012]

Miyamoto, T. and Yamaguchi, S., 2008. MceSim: A Multi-car Elevator Simulator, *IEICE Transactions on Fundamentals*, vol. E91-A, pp. 3207-3214.

Onat, A. et al. 2010. Design and Implementation of a Linear Motor for Multicar Elevators, *IEEE Transactions on Mechatronics*, vol. 15, pp. 685-693.

Valdivielso, A. and Miyamoto, T. 2011. Multicar-elevator Group Control Algorithm for Interference Prevention and Optimal Call Allocation, *IEEE Transactions on Systems, Man, and Cybernetics, Part A*, vol. 41, pp. 311-322.

Tanaka, S. and Watanabe, M. 2009. Optimization-based Collision Avoidance in Multi-car Elevator Systems, *Proceedings of ICCAS-SICE 2009*, pp. 764-769, August 18-21, Fukuoka (Japan).

Tanaka, S. and Watanabe, M. 2010. Improvement of the Optimization-based Collision Avoidance Method for Reversal- and Livelock-free Operation in Multi-car Elevator Systems, *Proceedings of SICE 2010*, pp. 844-848, August 18-21, Taipei.

Tanaka, S. and Miyoshi, T. 2011. Performance Comparison between Immediate and Nonimmediate Passenger Guidance in Group Control for a Multi-car Elevator System, *The 13rd International Conference on Harbor, Maritime & Multimodal Logistics Modeling and Simulation (HMS2011)*, pp. 130-135. September 12-14, Rome (Italy).

Strakosch, G.A. (Ed.), 1998. *The Vertical Transportation Handbook*. 3rd ed., John Wiley & Sons, Inc.