A Problem and Its Solution for Multi-Car Elevator Group Control

Yuki Kuroda[1], Mitsuru Nakata[2], Shingo Yamaguchi[1]
[1]Graduate School of Science and Engineering, Yamaguchi University,
16-1 Tokiwadai 2-chome, Ube-shi, 755-8611 Japan
[2]Faculty of Education, Yamaguchi University
1677-1 Yoshida, Yamaguchi-shi, 753-8513 Japan
[1]Tel: +81-836-85-9510 Fax: +81-836-85-9501
[2]Tel: +81-83-933-5402 Fax: +81-83-933-5304
E-mail : [1]{m008vk, shingo}@yamaguchi-u.ac.jp, [2]mnakata@yamaguchi-u.ac.jp

Abstract: Multi-car elevator (MCE for short) systems have been receiving increasing attention these days. An MCE has two or more cars in one shaft. It is difficult to control MCEs by using algorithms developed for single-car elevator systems, because MCEs need to avoid collision of the cars of the same shaft. In this paper, we propose an algorithm to control MCE systems, and show the effectiveness of our algorithm through computer simulation.

1. Introduction

Nowadays there are two or more elevators in a building. Those elevators are usually controlled as a group to increase the total transportation power. This is called *group control*. Several algorithms for group control have been proposed. Inamoto et al.[1] have applied a branch-and-bound method to group control. Eguchi et al.[2] have optimized a group controller by genetic network programming.

In addition to single-car elevators (SCE for short), new elevators has been proposed such as double-deck and multi-car. Multi-car elevator (MCE for short) systems have been receiving increasing attention these days. An MCE has two or more cars in one shaft. It is difficult to control MCEs by using algorithms developed for SCEs, because MCEs need to avoid collision of the cars of the same shaft.

A few algorithms have been proposed for MCE systems. Suzuki et al.[3] have applied genetic algorithm to the optimization of control rules for MCEs. Shiraishi et al.[4] have proposed an adaptive control technique for MCEs by adopting learning automatons. Ikeda et al.[5] have applied evolutionary multi-objective optimization to design of an MCE controller.

In this paper, we formalize group control for MCEs as a problem, and give a solution of this problem, i.e. an algorithm for MCE group control.

2. Problem

An MCE system (see Fig. 1) is composed of the following.

- *Cars*
- *Shafts*: A car is fixed to one shaft, and cannot pass another car.
- *Floors*: The first floor, which has the doorway of the system, is called the *terminal floor*. Floors higher than the terminal floor are called *general floors*. A general floor with more hall calls than the others is called a *special floor*. Floors lower than the terminal floor are called *garage floors*, which enable all cars to process hall calls to/from the terminal floor.

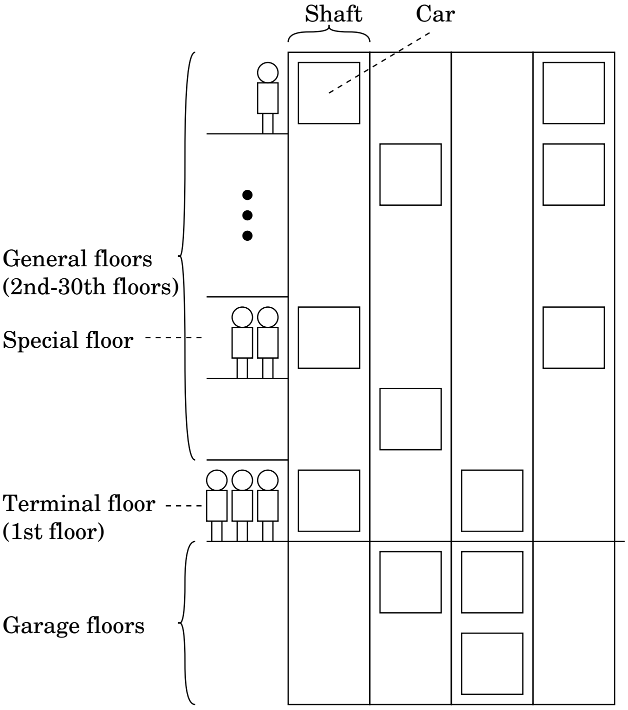

Figure 1. An example of MCE systems, which has 4 shafts, 3 cars per shaft, and 30 floors, i.e. $4S3C30F$.

The MCE system with x shafts, y cars per shaft, and z floors is denoted by $xSyCzF$. $xSyCzF$ has $(y-1)$ garage floors. Figure 1 shows $4S3C30F$.

A hall call occurs when persons call a car. If n persons call a car to move from the s-th floor to the d-th floor at time t, its hall call is denoted by (t, s, d, n). The s-th and d-th floors are called *start* and *destination* floors, respectively. Hall calls statistically occur according to the following traffic patterns.

- *Up-peak*: The hall calls from the terminal floor account for 90% of all hall calls.
- *Down-peak*: The hall calls to the terminal floor account for 90% of all hall calls.
- *Special floor*: The hall calls from/to the special floors account for 90% of all hall calls.
- *Ordinary*: Hall calls occur randomly.

The controller of MCE $xSyCzF$ is composed of one *group controller*, $(x \times y)$ *car controllers* and x *shaft con-*

trollers. The group controller assigns a hall call to a car controller. According to this assignment, the car controller drives the car in charge. The group controller can also give the following commands to a car controller in order to improve the total performance.

- *Assignment of a pseudo call*: The car controller immediately move the car in charge independent of the assigned hall calls.
- *Prolongation of stop time*: The car controller keeps the car stop.

A shaft controller checks the possibility of collision among the cars in charge. If there is the possibility of collision, the shaft controller orders the car controllers to prevent the collision.

We formalize MCE group control as the following problem.

Definition 1 (MCE group control problem) Minimize the average *service completion time* for the hall calls occurring between 1 [sec] to 5400 [sec] in simulation subject to the following.

- Events occur at time $1, 2, 3, \cdots$ [sec].
- The specification of MCE $xSyCzF$ is as follows: x=$3, 4, \cdots, 8$, y=$2, 3, \cdots, 5$, and z=$10, 11, \cdots, 50$.
- The scheduling policy of car controllers is as follows.
  - A car moves from the start floor to the destination floor via the shortest path, except the detours by pseudo calls.
  - A car skips the floor which persons only ride in if the car is a capacity crowd. □

This problem was used in the CST Solution Competition 2007 [6]. For the detail of the problem, refer to the web site of the competition [7].

3. Algorithm

We proposed an algorithm to solve the above problem, which was the most excellent of ones proposed in the CST Solution Competition 2007.

The basic policy of our algorithm is to prioritize avoidance of car collision. Our algorithm is based on two restrictions. The first restriction is called *zone restriction* [8], [9]. This restriction is used to assign a hall call to a car dealing with a zone which includes higher floor among the start floor and destination floor of the hall call. The second restriction is called *direction restriction*. There are two kinds of car collisions: the *head-on* collision and the *rear-end* collision. The head-on collision can be avoided by synchronizing direction of all car moving. On the other hand, the easiest method of avoiding the rear-end collision is to stop backward cars until rear-end collisions disappear most probably. The outline of our algorithm is shown in Fig. 2 and in the following.

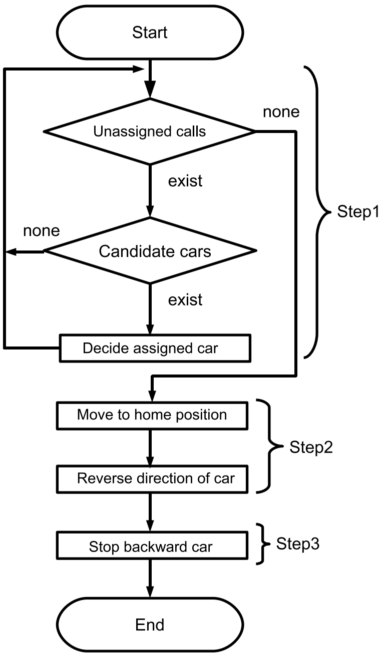

Figure 2. The flow chart of our algorithm

Step 1: Assign hall calls to cars according to the following.

1-1° Select candidate cars for a hall call according to zone restriction and direction restriction.

1-2° Decide a car to assign from candidate cars according to the evaluated value, which will be described later.

Step 2: Move cars which have not any assignments to each home position and reverse directions of the cars. The home position of each car is decided according to its zone.

Step 3: Stop backward cars to avoid the collision if rear-end collision likely occurs in the future.

In 1-1°, candidate cars are selected based on the two restrictions. The second restriction is based on two concepts: *call-direction* and *car-direction*. The call-direction of a hall call means the direction from the start floor to the destination floor. The car-direction of a car means the movable direction. Under this second restriction, a hall call is assigned to a car only when the call-direction is the same as the car-direction. As a result, a head-on collision is completely excluded in this way.

In 1-2°, a car is selected from candidate cars according to 1-1° and a hall call is assigned to the car. The selection is based on an evaluated value evl_k of a car car_k. A hall call is assigned to the car with the minimum evaluated value. car_k is calculated according to the following expression.

$$evl_k = Wait(C_k \cup \{x\}) - Wait(C_k)$$

Here, C_k is a set of hall calls which have been assigned to car_k but not been served yet. Moreover, $Wait(C_k)$ is the total estimated waiting time of car_k and x is a hall call which is ready to be assigned to car_k. The evaluated value evl_k shows delay of the total estimated waiting time of car_k when call x is assigned to car_k. If evl_k is small enough, then the assignment of x to car_k would give little influence on waiting time of car_k.

In step 2, cars, to which no hall calls have been assigned, are moved to each home position first.

When direction of a car is Up, its home position is the bottom floor of this car's zone. And the car is moved to its home position if a virtual current floor of the car is more below than the home position. Note that the virtual current floor means the nearest floor where the car can stop. When direction of a car is $Down$, the car is moved to its garage floor (or a terminal floor if the car is the highest). In other words, the home position is the garage floor (or the terminal floor).

Next, directions of the cars are reversed according to the following. Because all of car-direction in a shaft are synchronized based on the direction restriction, all of car-directions have to be reversed before assigning calls which have the counter direction to the cars.

Reversing from Up to $Down$: When the following three conditions are satisfied, car-direction of car_k is reversed from Up to $Down$: (i) car$_k$ exists at above than the bottom floor of car$_k$'s zone; (ii) the car has not been assigned any calls; (iii) direction of its right under car is $Down$.

Reversing from $Down$ to Up: When the highest car in a shaft stays at the terminal floor, car-direction of all car are reversed from $Down$ to Up.

In step 3, backward cars are stopped if rear-end collisions may probably occur in the future. This technique is called *safety restriction*. The safety restriction is to forecast the possibility of collisions and to make backward cars stopping until the possibility of collisions disappears. In other words, this restriction makes car stopping until the safety on the moving direction of the car is confirmed completely. Specifically, when a car begins moving or is moving, its safety restriction at the next time is confirmed. If the condition of the next time isn't safe, then the car is stopped. For the safety confirmation, the next stopping floor of a certain car (the marked car at the following) and virtual current floor of a car which may collide with the marked car are compared. If the next stopping floor of a marked car goes ahead of the virtual current floor of an anterior car, the marked car is stopped by assigned pseudo call. When safety of car-direction is confirmed, the car begins to move again.

Our algorithm is especially effective at Up-peak, because the highest car in a shaft is moved to the terminal floor whenever car-direction is reversed from $Down$ to Up.

4. Simulation Results

To evaluate our algorithm, we have compared the average service completion time of MCE and SCE under Up-peak, Down-peak, Special floor and Ordinary traffic patterns. The MCEs ($3S2C30F$ and $3S3C30F$) adopt our algorithm, and the SCEs ($5S1C30F$, $6S1C30F$, and $9S1C30F$) adopt assignment of selection car at random.

Figure 3 shows the results in Up-peak. In the figure, the horizontal axis and the vertical axis mean the rate of occurring hall calls (hall call rate, for short) and the average service completion time, respectively. MCE $3S2C$ which is applied our algorithm to gets worse drastically on hall call rate about 80. We call this rate turning rate. Figure 4 shows the results in Down-peak. Turning rate of MCE $3S2C$ is about 60. Figure 6 shows the results in Ordinary. Turning rate of MCE $3S2C$ is about 80. Our algorithm is more efficient in Up-peak and Ordinary than in Down-peak from these results.

Next, we compared MCE $3S2C$ with MCE $3S3C$. In Up-peak, there is not turning rate of MCE $3S3C$ within 100. In Down-peak, turning rate of MCE $3S3C$ is about 90. In Ordinary, there is not turning rate of MCE $3S3C$ within 100. If we increase cars in MCE, MCE's performance is improved generally.

We compared the MCEs with SCEs. In Up-Peak, MCE $3S2C$ is more efficient than $5S1C$, and $3S3C$ is much better than $6S1C$ when hall call rate is large. $3S3C$ does not reach $9S1C$ although $3S3C$ has the same number of cars as $9S1C$. In Down-peak, $3S2C$ is not over $5S1C$ though hall call rate is large. As well as $3S3C$ is not over $6S1C$. So, in Down-peak, our algorithm of MCE is not as effective as in Up-peak. Figure 5 shows results in Special Floor. Special floor is set 20 floor. MCEs are inferior than SCEs. Both of $3S2C$ and $3S3C$ don't reach $5S1C$. This result is the worst among 4 call patterns. So, our algorithm is incompatible with Special Floor. In Ordinary, $3S2C$ reaches between $5S1C$ and $6S1C$ when hall call rate is large. And $3S3C$ is better than $6S1C$ finally.

Our evaluation results show the following.

- Our algorithm is effective in Up-peak, Ordinary.
- The MCEs which are applied our algorithm to are more efficient than the SCEs when hall call rate is large.
- If we increase the number of cars in MCE, MCE's transportations are improved generally.

In any call pattern, $3S3C$ does not reach $9S1C$ although $3S3C$ has the car of the same number as $9S1C$. In Up-Peak and Ordinary, $3S2C$ is over than $5S1C$, and $3S3C$ is over than $6S1C$ when hall call rate is large. In Down-Peak, $3S3C$ catch up with $6S1C$ finally. In these cases, we cut down the number of shafts with remaining performance by using our algorithm.

Our algorithm is not efficient when hall call rate is small. When car is not assigned hall calls to, we move cars to home position for avoidance of the collision in future. But, this action may adversely affect performance. Because, if avoidance of the collision don't occur, moving cars to home position will be wasted action. So, if we move cars to home position only at need, MCE's performance may be improved. Under our algorithm, cars in one shaft move only at the same direction for avoidance of the head-on collision. If a car faces another but the distance between them is long, the collision may not occur. This restriction may adversely affect performance. If we adopt more flexible restriction rules, MCE's performance will be better.

5. Conclusion

In this paper, we have formalized MCE group control as a problem, and have proposed an algorithm as a solution of this problem.

The basic policy of our algorithm is to prioritize avoid-

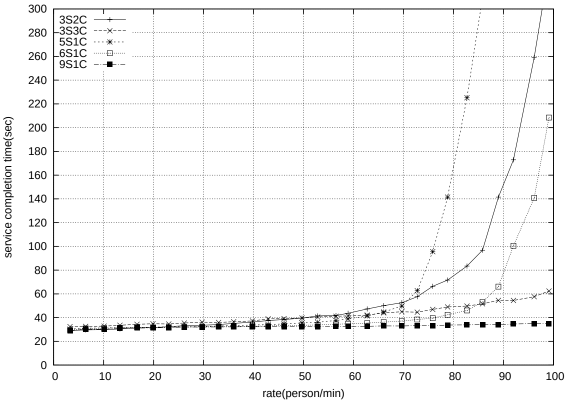

Figure 3. 30 floor. Up-peak.

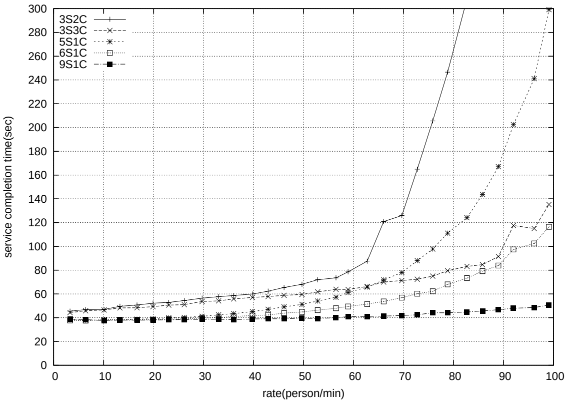

Figure 4. 30 floor. Down-peak.

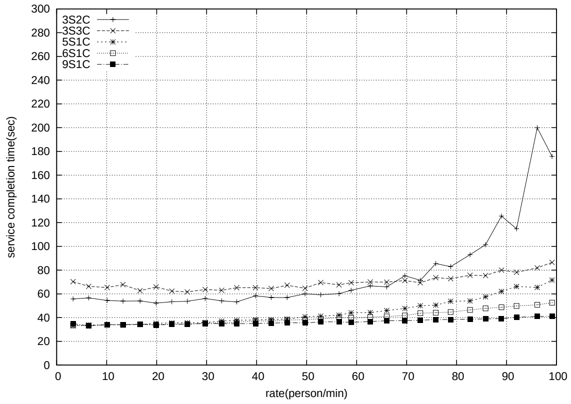

Figure 5. 30 floor. SpecialFloor(20).

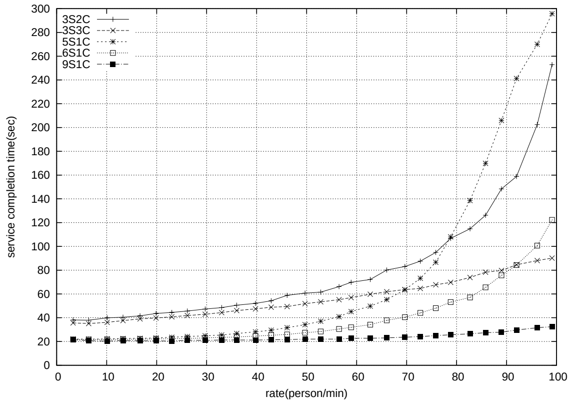

Figure 6. 30 floor. Ordinary.

ance of car collision. Hall calls are assigned to cars by two restriction which are the zone restriction and the direction restriction. We have evaluated our algorithm of MCE group controller by comparing with SCE and considered feature of our algorithm. Our algorithm prioritize avoidance of car collision. On the other hand, performance is not good when hall call rate is small. This reason is that the restrictions are too hard. As the future works, we need to improve performance when hall call rate is light, and adopt more flexible restrictions.

Acknowledgements The authors would like to thank Associate Professor Toshiyuki Miyamoto (Osaka Univ.) and Mr. Hiromichi Suzuki and Mr. Ken'ichi Aoki (Fujitec Co., Ltd.) for contribution on formalization of the problem.

References

[1] T. Inamoto, H. Tamaki, H. Murao, S. Kitamura, "Deterministic optimization model of elevator operation problems and an application of branch-and-bound method", IEEJ Trans. EIS, vol.123, no.7, 1334-1340, 2003.

[2] T. Eguchi, K. Hirasawa, S. Markon, "Optimization of elevator group supervisory control system by genetic network programing", The Japan Society of Mechanical Engineers vol.2005, pp.17-20, 2005.

[3] H. Suzuki, S. Takahashi, Y. Sano, T. Sudo, S. Markon, H. Kita, "Simulation-based optimization of multi-car elevator controllers using a genetic algorithm", T. SICE, vol.40, no.4, pp.466-473, 2004.

[4] K. Shiraishi, T. Hamagami, H. Hirata, "Multi car elevator control by using learning automaton," IEEJ Transactions on Industry Applications, vol.125, no.1, pp.91-98, 2005.

[5] K. Ikeda, H. Suzuki, S. Markon, and H. Kita, "Designing traffic-sensitive controllers for multi-car elevators through evolutionary multi-objective optimization," Proc. of 4-th Intl. Conf. on Evolutionary Multi-Criterion Optimization, LNCS vol.4403, pp.673-686, 2007.

[6] CST Solution Competition 2007, `http://www.ieice.org/~cst/comp07/`

[7] Detail of CST Solution Competition 2007, `http://www.ieice.org/~cst/comp07/compe07problemDetails3.pdf`

[8] H. Suzuki, "A study on intra-shaft operating method for multi-car elevators," IEICE Technical Report CST2007-4, pp.19-24, 2007.

[9] S. Yamashita, M. Iwata, S. Hikita, "A study on operating method of double-car elevator system," Proc. of Recent technology and advancement on elevators and play facilities, pp.25-28, 2004.