Multi-Car Elevator Group Control: Schedule Completion Time Optimization Algorithm with Synchronized Schedule Direction and Service Zone Coverage Oriented Parking Strategies

Alex Valdivielso, Toshiyuki Miyamoto and Sadatoshi Kumagai
Graduate School of Engineering, Osaka University, Japan
Yamadaoka 2-1, Suitashi, Osaka, 565-0871, Japan
Tel: +06-6877-3564, Fax: +06-6879-7263
E-mail: alex@is.eei.eng.osaka-u.ac.jp,{miyamoto, kumagai}@eei.eng.osaka-u.ac.jp

Abstract: Multi-car elevator systems consist of shafts with two or more independent cars operating in them. Due to their special characteristics, such as the need to avoid the interference among cars in the same shaft, conventional control methods cannot be implemented on them. Therefore it is necessary to develop group control methods for multi-car elevator systems able to perform an optimal floor call allocation while considering their special characteristics. In this paper we propose a group control method for multi-car elevator systems consisting of a schedule completion time optimization algorithm, an interference risk prevention strategy using the idea of synchronized schedule direction, and parking strategies to assure the coverage of service zones according to the identified passenger traffic pattern. In order to test its performance the proposed algorithm was compared with a sample zoning algorithm in the presence of different passenger traffic patterns. According to the simulation results the performance of our algorithm excels the performance of the sample zoning algorithm in the inter-floor and down-peak traffic patterns. In the case of the up-peak traffic pattern, for which the sample zoning algorithm presents its best performance, the proposed algorithm shows a better result under the arrival rate of 90 passengers/min.

1. Introduction

The concept multi-car elevator has been commonly used to name elevator systems containing more than one single-car shaft. However, as in [1], [2], [3], in this paper we will define the term Multi-Car Elevator (MCE) as an elevator system having two or more cars built in the same shaft. Although extensive research has been performed on the group control of elevator systems, control techniques developed for conventional elevators cannot be easily applied in MCE systems due to their inherent characteristics, such as their need to prevent interference between cars operating in the same shaft. Therefore it is necessary to develop a group control method for MCE systems that can efficiently perform the allocation of floor calls while considering their inherent characteristics. In this paper we propose a MCE group control method consisting of a floor call allocation strategy, which is composed by a car interference prevention strategy and a schedule completion time optimization algorithm; and a parking strategy to assure the coverage of service zones according to the identified passenger traffic pattern.

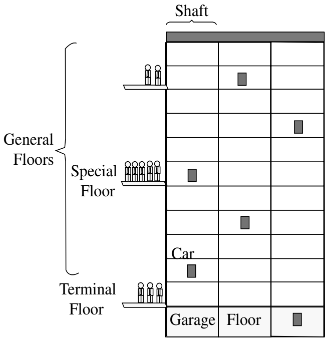

Figure 1. MCE System

1.1 MCE System

As described by [3] there are two main types of MCE systems, the MCE system that allows horizontal displacement and the MCE system that allows only vertical displacement. In this paper we deal just with the latter one. In other words, the condition of cars in the same shaft not being able to surpass each other traveling up or down is assumed. In this paper, just as in [3] the elevator call and the destination entry are assumed to be done simultaneously. In other words, the hall call and the car call described in [4] are combined into what [2] describes as floor call.

1.1.1 MCE Structure

A complete description of the elements that compose an MCE system is provided by [3]. To describe the MCE system configuration used we use the notation $x\mathrm{S}y\mathrm{C}z\mathrm{F}$ proposed by [2] in which x denotes the number of elevator shafts, y the number of cars built in each shaft, and z the number floors the building consists of. Figure 1 shows a 3S2C11F MCE configuration with its elements.

2. Group Control Algorithm

The proposed algorithm consists mainly of a floor call allocation strategy and a parking strategy. We refer to this Schedule Completion time optimization algorithm with Synchronized schedule Direction and Service zone coverage oriented Parking strategies as the SCSDSP algorithm. The control flow of the algorithm is shown in Fig. 2. The following sections provide a full description of the floor allocation and the parking strategies used.

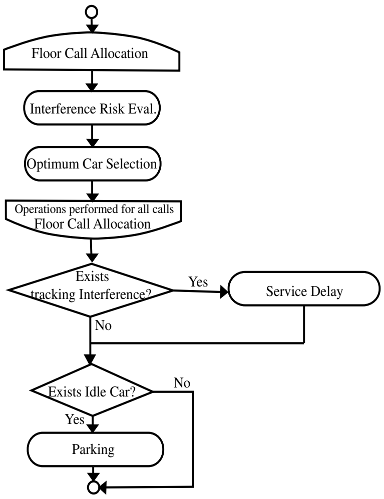

Figure 2. Algorithm Flow

2.1 Terminology

Before presenting the strategies that compose the SCSDSP algorithm, we introduce the following terms:

- c_{ij}: From bottom position to top the i-th car of shaft j, where $i = 1, ..., y$ and $j = 1, ..., x$ in a xSyCzF MCE structure
- v_{ij}: Virtual floor of car c_{ij}. Virtual floor is defined as the current floor in which the car is located when parked or the floor where it could be able to stop, if indicated, when moving at a certain velocity [2].
- F_{ij}: Set of floors contained in the schedule of car c_{ij}
- r: Unassigned floor call
- A_{ij}: Set of calls assigned to c_{ij}
- s_r: Departure floor of floor call r
- d_r: Destination floor of floor call r

2.2 Floor Call Allocation

When a floor call is generated each car is evaluated to verify if, by responding to this call, it might interfere with the operation of another car running in the same shaft. Only cars having no interference risk detected are included in the candidate list of cars that can respond to the current floor call. Then for each car in the candidate list the schedule completion time S is calculated. The car c_{ij} presenting the minimum S_{ij} is chosen as the optimum car to respond to the current floor call. Whenever there is no candidate to respond, the floor call is stored to be re-evaluated in the next simulation cycle. This is done until the floor call is assigned to a car.

2.2.1 Interference Risk Evaluation

Before presenting the interference prevention conditions that compose the Interference Risk Evaluation function, it is necessary to define the call direction of a given floor call r (dir_r) and the schedule direction of car c_{ij} ($schedDir_{ij}$) :

- $dir_r = \begin{cases} Up & d_r > s_r \\ Down & d_r < s_r \end{cases}$
- $schedDir_{ij} = \begin{cases} Up & \text{if} & \forall a \in A_{ij}, d_a > s_a \\ Down & \text{if} & \forall a \in A_{ij}, d_a < s_a \\ None & \text{if} & A_{ij} = \emptyset \\ Bidirectional & \text{otherwise} \end{cases}$

In addition, we denote the moving direction of c_{ij} as $movDir_{ij} = \{Up, Down, Stay\}$ [2].

In MCE systems basically three types of interferences arise, the interference between cars traveling in opposite directions (opposing interference), the interference between cars traveling in the same direction (tracking interference), and the interference caused by idle cars (impeding interference). The Interference Risk Evaluation function deals only with the opposing interference type. The prevention of the tracking interference type and the impeding interference type is explained in section 2.3 and in section 2.4.1 respectively. In order to prevent the occurrence of opposing interference, not only in the present time but also in a future point in time, the schedule direction of cars is taken as the base of the interference risk evaluation. The first step is to make the schedule of each car to contain only calls having the same direction. This is done by applying the following principles (See ESA in [5]).

- The schedule direction of a car is determined by the first call it is assigned while in idle state.
- A car will not respond to calls having direction opposite to its schedule direction.
- Cars with schedule direction = *None* can take any call as long as they satisfy the interference prevention conditions.

By following these principles the schedule direction = *Bidirectional* case will never occur, and we base our interference prevention conditons on the cases of schedule direction = $\{$*Up, Down, None*$\}$. In a xSyCzF Mce configuration c_{ij} will become a candidate to respond to r if the following conditions are satisfied:

- case: $schedDir_{ij} = Up$
  - ◇ $(s_r \geq v_{ij}) \wedge (schedDir_{i+1,j} \neq Down)$
  - ◇ $(movDir_{ij} \neq Down)$
- case: $schedDir_{ij} = Down$
  - ◇ $(s_r \leq v_{ij}) \wedge (schedDir_{i-1,j} \neq Up)$
  - ◇ $(movDir_{ij} \neq Up)$
- case: $(schedDir_{ij} = None) \wedge (dir_r = Up)$
  - ◇ $(schedDir_{i+1,j} \neq Down) \vee ((schedDir_{i+1,j} = Down) \wedge (W(c_{i+1,j}, d_r) > S_{ij}))$
  - ◇ $(s_r \geq v_{ij}) \vee ((s_r < v_{ij}) \wedge (c_{ij} = c_{yj}) \wedge (W(c_{i-1,j}, s_r) > W(c_{y,j}, s_r)))$
- case: $(schedDir_{ij} = None) \wedge (dir_r = Down)$
  - ◇ $(schedDir_{i-1,j} \neq Up) \vee ((schedDir_{i-1,j} = Up) \wedge (W(c_{i-1,j}, d_r) > S_{ij}))$
  - ◇ $(s_r \leq v_{ij}) \vee ((s_r > v_{ij}) \wedge (c_{ij} = c_{yj}))$

Function $W(c_{ij}, f_l)$ represents the arrival time of car c_{ij} to floor f_l. The arrival time calculation considers the floors contained in F_{ij} in which c_{ij} will stop before reaching f_l. Consider $f_k \in F_{ij}$ as the next floor in the schedule of c_{ij}. Where $k = 1, ..., m$ is the floor order index for F_{ij} containing a total of m floors. We define the time required by c_{ij} to reach from f_{k-1} to f_k considering its current state x_{ij} as $t(f_{k-1}, f_k, x_{ij})$ [4]. The current state x_{ij} refers to the current elevation of c_{ij}

and its current velocity or if parked, to the time it must remain in that floor before departing to the next scheduled floor. The information of x_{ij} is only used to calculate travel time from f_0 to f_1. In addition, the time to board and descend passengers t_b is also considered. The arrival time $W(c_{ij}, f_l)$ is defined by equation (1).

$$W(c_{ij}, f_l) = \sum_{k=1}^{l}(t(f_{k-1}, f_k, x_{ij}) + t_b) \qquad (1)$$

2.2.2 Optimum Car Selection

The selection of the optimum car is done through the optimization of the schedule completion time. First we take the time function $t(f_{k-1}, f_k, x_{ij})$ defined for the $W(c_{ij}, f_l)$. In addition we consider t_b as the time required in order to board or descend passengers at a floor. Finally, we also take into account the total time delay it must be applied to c_{ij} in order to prevent tracking interference when responding to the current call (See section 2.3). We denote this time delay as $t_{penalty}$. The schedule completion time function S_{ij} of an elevator car c_{ij} is defined then as the total time required by c_{ij} to reach and stop to board or descend passenger at every floor in its schedule, starting from v_{ij}. The schedule completion time function is described by the following equation:

$$S_{ij} = \sum_{k=1}^{m}(t(f_{k-1}, f_k, x_{ij}) + t_b) + t_{penalty} \qquad (2)$$

To calculate $t_{penalty}$ first we introduce the set of scheduled floors of $c_{i+1,j}$ including d_r, $F'_{i+1,j} = F_{i+1,j} \cup \{d_r\}$. If we assume d_r will be the n-th floor to be reached by $c_{i+1,j}$, in other words $f_n = d_r$, we can define $G = \{f_k \in F'_{i+1,j} \mid k < n\}$ as the set of floors in which $c_{i+1,j}$ has to stop before surpassing the elevation of the destination floor of the current call. In the same way we define $H = \{f_k \in F'_{i-1,j} \mid k > n\}$ for $c_{i-1,j}$. Thus $t_{penalty}$ is calculated as shown by equation (3).

$$t_{penalty} = (\mid G \mid + \mid H \mid)t_b \qquad (3)$$

The selection of the optimum car is done following the procedure described below:

I. Add s_r and d_r to the schedule of each candidate car, F_{ij}, tentatively.
II. Calculate S_{ij} for each candidate car
III. $c_{opt} = \arg\min_{ij} S_{ij}$

2.3 Service Delay

The tracking interference type occurs when c_{ij} is assigned to respond to call r with direction either Up or Down and it is not able to do it because it is forced to wait for $c_{i+1,j}$ ($dir_r = Up$) or $c_{i-1,j}$ ($dir_r = Down$) to stop at each of its scheduled floors located before d_r. In order to prevent this, after the floor call allocation process the optimum car is evaluated and if tracking interference risk is detected it is forced to delay its response to the floor call by extending its parking time at its virtual floor. The total time c_{ij} will delay its response to r is equal to $t_{penalty}$ which is calculated in (3) (See Fig. 2).

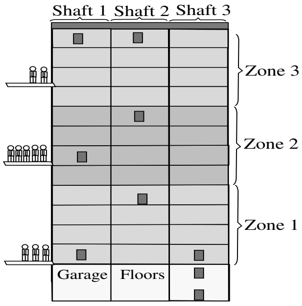

Figure 3. Parking zones for a 3S3C12F MCE System

2.4 Parking

The parking strategies described below are applied only to idle cars as described in Fig. 2. In this research the parking of cars pursues mainly two objectives, to help prevent the occurrence of impeding interference between cars operating in the same shaft and to try to balance the distribution of floor calls among all cars in the system. The impeding interference prevention is done by the *Interference Prevention Parking* function; while the balance of the floor call allocation is done by dividing the building into service zones according to the number of cars built per shaft. Cars are assigned to park in one or several zones depending on the passenger traffic pattern identified (See Fig. 3). It is important to mention that the *pseudo calls* [2] generated to park cars are also evaluated to prevent risk interference before being assigned.

2.4.1 Interference Prevention Parking

Whenever an idle car becomes an obstacle for a car moving upward or downward, it is forced to move to one floor above or below the next stop floor of the operating car for which it is an obstacle. While a car is being parked to prevent impeding interference it cannot have any of the following parking strategies applied.

2.4.2 Inter-floor Strategy

During the inter-floor traffic pattern calls are expected to be generated from any service floor in any direction. Therefore, we seek to balance the coverage of the service floors by having the idle upper cars $(c_{2j}, ...c_{yj})$ park at the highest floor of its assigned zone [3]; while the idle bottom car (c_{1j}) is parked at the lowest floor of Zone 1, in other words at the terminal floor. This is shown in Fig. 3 in Shaft 1.

2.4.3 Down-peak Strategy

During the down-peak traffic pattern it is expected to have 90% of the floor calls generated from higher floors with the terminal floor as destination. During this pattern we concentrate the idle cars in the higher section of the building by parking them in the highest floor of their assigned service zone. This is shown in Fig. 3 in Shaft 2. To prevent the upper cars $(c_{2j}, ...c_{yj})$ from monopolizing the generated calls the following condition is introduced when facing this passenger traffic pattern:

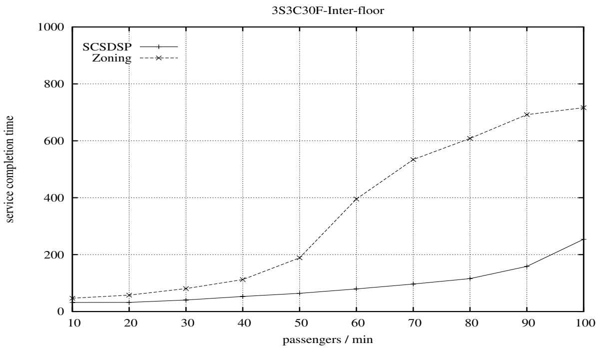

Figure 4. Comparison of the performance of SCSDSP and a Zoning algorithm in inter-floor traffic

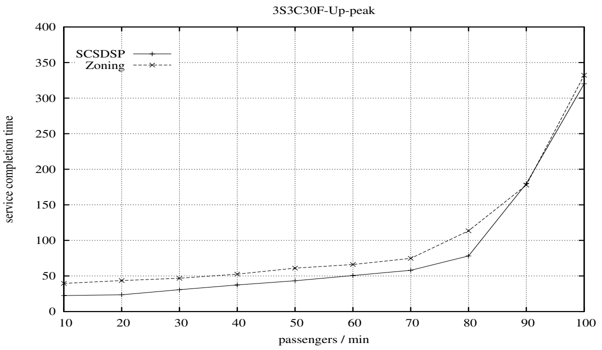

Figure 5. Comparison of the performance of SCSDSP and a Zoning algorithm in up-peak traffic

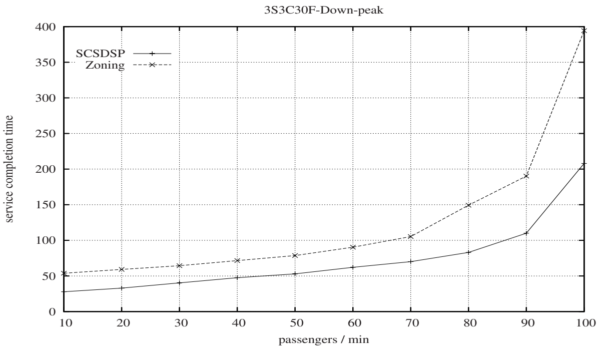

Figure 6. Comparison of the performance of SCSDSP and a Zoning algorithm in down-peak traffic

- Let P_i be the set of floors of the zone assigned to c_{ij} and $p_{lowest} \in P_i$ the lowest of the floors that compose this zone. Car c_{ij} will not respond to r if $s_r < p_{lowest}$ when $schedDir_{ij} = None$

2.4.4 Up-peak Strategy

During the up-peak traffic pattern it is expected to have 90$\%$ of the calls with the terminal floor as departure floor and as destination higher floors. Therefore, we concentrate the service coverage on the terminal floor by assigning the idle top car (c_{yj}) to park at the terminal floor while the idle lower cars $(c_{y-1,j}, ..., c_{1,j})$ are assigned to return and park at their respective *garage floor* [3]. This is shown in Fig. 3 in Shaft 3.

3. Simulation Results

The simulation was performed using the MceSim simulator [6]. For the simulation a 3S3C30F MCE configuration, the arrival rate range was 10-100 passengers/min, and the simulation time 5400s. Figures 4 - 6 show a comparison between the performance of the SCSDSP algorithm and a sample zoning algorithm [6] in different passenger traffic patterns. From the results it can be seen how the performance of the SCSDSP algorithm excels the performance of the zoning algorithm in the presence of inter-floor and down-peak traffic patterns. In the up-peak traffic pattern, for which the zoning algorithm has its best result, the SCSDSP algorithm shows a better performance below the 90 passengers/min arrival rate.

4. Conclusions

In this paper we proposed, a group control method for MCE systems which consists mainly of a floor call allocation strategy and car parking strategy. The floor call allocation depends on the *Interference Risk Evaluation* function and the optimization of the schedule completion time; while the parking applies a specific strategy for each passenger traffic pattern. In addition through the simulation results it was proven that the SCSDSP algorithm shows a better performance than a zoning algorithm in the presence of inter-floor and down-peak traffic patterns. In the case of the up-peak traffic pattern, the proposed SCSDSP algorithm shows a better performance under the 90 passengers/min arrival rate.

References

[1] J. Gale, “ThyssenKrupp’s TWIN Lift System”, Elevator World, vol.51, no.7, pp.51-53, 2003.

[2] CST Solution Competition 2007. Problem Details, http://www.ieice.org/˜ cst/compe07/ compe07problemDetails2.pdf

[3] H. Suzuki, S. Takashi, Y. Sano, T. Sudo, S. Markon, and H. Kita, “Simulation-based Optimization of Multi-car Elevator Controllers Using a Genetic Algorithm”, SICE,Vol. 40, No. 4, pp.466-473, 2004.

[4] K. S. Wesselowski and C. G. Cassandras, “The Elevator Dispatching Problem: Hybrid System Modeling and Receding Horizon Control”,Proc. of 2nd IFAC Conference on Analysis and Design of Hybrid Systems, pp. 136-141, 2006.

[5] G. Bao, C. G. Cassandras, T. E. Djaferis, A. D. Gandhi, and D. P. Looze, “Elevator Dispatchers for Down-Peak Traffic”, Technical Report, University of Massachusetts,Department of Electrical and Computer Engineering, 1994.

[6] T. Miyamoto and S. Yamaguchi, “CST Solution Competition 2007 Detail of Evaluation Tool”, IECIE Technical Report CST2007-26, pp.13-18, 2007.