Estimated Time of Arrival (ETA) Based Elevator Group Control Algorithm with More Accurate Estimation

Aiying Rong
University of Turku, Department of Information Technology,
Lemminkäisenkatu 14 A, FIN-20520 Turku, Finland

Henri Hakonen
University of Turku, Department of Information Technology,
Lemminkäisenkatu 14 A, FIN-20520 Turku, Finland

Risto Lahdelma
University of Turku, Department of Information Technology,
Lemminkäisenkatu 14 A, FIN-20520 Turku, Finland

Turku Centre for Computer Science
TUCS Technical Report No 584
December 2003
ISBN 952-12-1289-6
ISSN 1239-1891

Abstract

We develop ETA (Estimated Time of Arrival) based elevator group control algorithms with more accurate estimations to minimize the average waiting time of the passengers. Following the principle of ETA estimations, the algorithms not only estimate the attending time of the new hall call, but also the delay that serving it will cause to successive unattended passengers that have been allocated to the same elevator. To increase the accuracy of this estimation we try to consider the number of extra stops caused by the new hall call and apply the three-passage concept to determine the service order of the hall calls: passage one (P1) hall calls are those that can be served by the elevator along its current travel direction, passage two (P2) hall calls require reversing the direction once, and passage three (P3) hall calls require two reversals.

We propose two variants of the algorithm: a basic variant and a reallocation variant. The basic variant is based on the immediate allocation policy. The reallocation variant is based on coordination between the basic variant and a heuristic reallocation mechanism. The time complexity of both algorithms is O(MN), where M is the number of the elevators and N is the number of floors in the building. We have performed test runs with traffic data generated from realistic buildings ranging from 9 to 40 floors and with 3 to 8 shafts for typical traffic patterns. Our basic ETA algorithm reduces the average waiting time by 16 % and reduces the percentage of passengers who wait for more than 60 seconds by more than 3 % points when compared with the ETA algorithm of the commercially available *Elevate* simulator. Our reallocation variant further reduces the average waiting time by 7 % and the percentage of the passengers who wait for more than 60 seconds by more than 2 % points as compared with our basic algorithm.

Keywords: Elevator Group Control, Estimated Time of Arrival, Three-Passage Concept, Immediate Allocation Policy.

1. Introduction

A large number of modern elevator systems consist of multiple elevators that transport the passengers to their destinations on the desired floors in the building. To make an elevator system operate safely and efficiently on passengers' behalf, some automatic controls are employed. The coordination of a group of elevators in the elevator systems for satisfying the demands of passengers is called the *elevator group control*. In the traditional up/down hall call button elevator systems, two events trigger the action of the elevator group controller. One is the hall call button on the floors and the other is the car call button inside the elevator. The basic function of the elevator group control is to assign appropriate elevators to occurred hall calls to quickly attend them. The usual performance criterion to be optimized is the average waiting time (AWT) of all the passengers in the system. The waiting time is the time period from the moment a passenger arrives to the moment when he boards some elevator.

The elevator group control problem is of inherently stochastic nature. The stream of arriving passengers is a stochastic process. Each passenger introduces three random variables: arrival time, arrival floor and desired destination floor. In addition, the number of passengers is also unknown behind each hall call in real elevator systems. All of these random sources must be considered when the decision is made about the appropriate elevator to serve the new hall call. Therefore, the elevator group control can be viewed as a combination of on-line scheduling, resource allocation and stochastic optimal control problem. The elevator group control problem has received extensive attention because of its theoretical importance and practical significance (Sutton & Barto 1998).

Two allocation policies are widely used in assigning elevators to serve hall calls: immediate allocation policy and continuous allocation policy. In the immediate allocation policy, new hall calls are allocated to an elevator as soon as they appear, remaining fixed once made. That is, the system tells the passenger immediately which elevator to use when the passenger makes a hall call. Japan typically adopts this policy and the destination-based control system (DCS) follows this assignment policy. In the continuous allocation policy, each elevator is allocated a maximum of one hall call, and some hall calls can remain unallocated. The unattended hall calls are continuously reevaluated to decide the appropriate elevator. This mode of the operation is typical of western countries.

The advantage of the immediate allocation is to shorten the psychological waiting time of the passengers. However, the early allocation may lose its optimality as the future traffic events change the situation. The advantage of continuous allocation is that the best elevator to serve a hall call can be updated as new calls are introduced into the system. However, elaborately designed continuous reallocation algorithms must in principle consider an exponential number of different calls-to-car assignments. Therefore, it is computationally expensive.

In this paper we develop Estimated Time of Arrival (ETA) based elevator group control algorithms with more accurate estimations to minimize the average waiting time of the passengers. Following the principle of ETA estimation, the algorithms not only estimate the attending time of the new hall call, but also the delay that serving it will cause to successive unattended passengers that have been allocated to the same elevator.

To increase the accuracy of ETA estimation we try to consider the number of extra stops caused by the new hall call and apply the three-passage concept (Cho et al. 1999, Gagov et al. 2001) to determine the service order of the hall calls: passage one (P1) hall calls are those that can be served by the elevator along its current travel direction, passage two (P2) hall calls require reversing the direction once, and passage three (P3) hall calls require two reversals.

We propose two variants of the enhanced ETA allocation algorithm. The basic variant adopts the immediate allocation policy. The reallocation variant is based on coordination between the basic variant and a heuristic reallocation mechanism. Each time only a finite number of carefully selected hall calls enter the reallocation cycle and they are reevaluated using the basic allocation algorithm. Therefore our reallocation variant is computationally efficient as compared with the continuous call allocation based group control algorithm.

The rest of this paper is organized as follows: Section 2 briefly reviews the group control algorithms from the viewpoints of dealing with the uncertainty in the system, Section 3 gives the simplified elevator system model, Section 4 presents the basic ETA estimation based on the three-passage concept, Section 5 discusses the coordination between basic ETA allocation algorithm and the reallocation mechanism, and Section 6 presents the results on test runs.

2. Review on elevator group control algorithms

The insurmountable combinatorial complexity and stochastic nature of the elevator group control problem has been attracting the attention of both researchers and practitioners to make their effort in this area. The elevator group control algorithms span from ad hoc approximations and heuristics to AI planning (Fujino et al. 1997, Kim et al. 1998, Siikonen 1997a) and decision-theoretic planning (Nikovski & Brand 2003a, 2003b).

The oldest elevator controllers used the principle of collective control in that elevators always stops at the nearest calls in their running direction. This kind of allocation strategy is not designed to deal with randomness. The drawback of this algorithm is *bunching*, where several elevators move close to each other and compete for the same hall call. This may result in the situation that elevators arrive at the same floor at about the same time. Generally only one elevator picks up the passengers and all other elevators waste time in making an extra stop, thus delaying the service time to other passengers.

Pure up-peak and down-peak traffic patterns can be analyzed from a theoretical viewpoint (Gamse & Newell 1982a, 1982b, Gandhi & Cassandras 1996). Special-purpose group control algorithms based on such traffic patterns have provably optimal solutions. The threshold based dynamic programming method (Pepyne & Cassandras 1997) provides an optimal solution during up-peak traffic. Dynamic zoning (Chan et al. 1997) can make the controller adapt to up-peak and down-peak traffic to obtain the optimal solution. However, since these kinds of algorithms were derived for special situations, they perform well during particular traffic patterns but fail to adapt to other traffic patterns.

To address the difficulty of the elevator group control problem, AI techniques have also found their way in the elevator area. Crites & Barto (1996) developed reinforcement learning (RL) algorithms that approximate dynamic programming (DP) on an incremental basis. It is computationally tractable in theory and guaranteed to converge to an optimal policy. However, the RL algorithm required 60,000 hours of simulated elevator operation to converge for one specific down-peak scenario. So it is not practical for real elevator systems. Genetic algorithms (GAs) represent the possible solutions to the problem by chromosomes. They use a fitness function to evaluate the performance criterion corresponding to individual chromosomes and improve the chromosomes by genetic operators such as selection, crossover and mutation. While GAs are good for finding high quality solutions, they are slow and inefficient in the use of computational resources. Therefore, GAs are difficult to use when the time available to achieve a solution is limited (Tyni & Ylinen 1999).

More rigorously motivated methods use approximations of the desired performance criterion that are computable in reasonable time. Estimated Time to Destination (ETD) and Estimated Time of Arrival (ETA) algorithms belong to this category. ETA and ETD attempt to treat each passenger in the system equally. When a new passenger is allocated to the elevator, the algorithms consider both the attending time of the new passenger and the delay that the new passenger will cause to the passengers that have already been allocated to this very elevator. The basic logic of allocation is same for ETA and ETD, but they use different optimization criteria. ETA minimizes the average waiting time of the passengers in the system while ETD minimizes the system time of the passengers in the system. Regular ETD and ETA address the situations without reallocation. Nikovski & Brand (2003a) developed a variant of the ETA algorithm based on the evaluation of average waiting time (AWT) by means of dynamic programming. However, they use the queue length information behind the hall call that would not be available in a real elevator system. Smith & Peters (2002) demonstrated the use of the ETD algorithm in the full DCS.

3. Simplified elevator system model

An elevator group control system consists of M elevators delivering passengers from their respective arrival floors to their destination floors in the N-floor building. The objective is to minimize the average waiting time of all the passengers in the system. The elevator system model is similar to that in (Hakonen, 2002).

The following traditional constraints on the behavior of the elevator are imposed:

(a) The elevator must not pass a floor at which a passenger wishes to get off;

(b) The elevator must fulfill its current commitment and not reverse direction with passengers in elevator;

(c) An empty elevator can stop, go up or go down.

The following assumptions are made throughout the paper.

(1) The arrival process follows the Poisson distribution.

(2) Both the acceleration and maximum speed of the elevator are constant. Thus, traveling time between any two floors can be fully determined by the distance between floors. The elapsed time since the elevator left immediate previous floor can be used to calculate the nearest floor at which the elevator can stop and the time to any floor. It is unnecessary for the controller to know the absolute position and velocity of the elevator explicitly. Therefore, the nearest floor and the travel direction are used to determine the elevator dynamics. If the elevator is not in motion, the nearest floor is its current floor.

(3) The bypass load of the elevator is 80 %. The moving elevator will not stop to respond a hall call when its load is equal to or greater than 80 % of its rated capacity.

(4) Each elevator follows the following service routes in attending the hall calls assigned to it: P1 hall calls → P2 hall calls → P3 hall calls, while the elevator capacity constraint is satisfied. If the elevator can not serve all of the passengers behind a hall call, the remaining passengers initiate a new hall call.

Next we will discuss details of our ETA based group control algorithms. To make the principle of ETA estimation clear, we introduce the number of passengers behind the hall call. However, we do not use this information in the algorithms.

4. Basic ETA estimation

Our basic ETA algorithm follows the principle of ETA estimation and is based on the three-passage concept, estimating the number of extra stops caused by the hall call besides the mandatory stops at occurred floor and the most likely reversal floor of the elevator.

4.1 Principle of ETA estimation

ETA attempts to treat each hall call (or passenger) equally by introducing the system degrade factor (SDF) to find the appropriate elevator to serve the new hall call. Not only is the attending time of the new hall call considered, but also the delay it will cause to the successive unattended hall calls that have been assigned to the same elevator. Thus, the total cost of allocating the new hall call to elevator *i* is given below.

$$t_i^{total} = \sum_{j=1}^{n_i} t_{i,j}^{delay} + t_i^{attending} \tag{1a}$$

Here n_i is the number of the passengers that have been assigned to elevator *i* and not been served yet, $t_{i,j}^{delay}$ (≥ 0) is the delay that the new hall call will cause to passenger *j*, who has been assigned to elevator *i* and not been served, $t_i^{attending}$ (≥ 0) is the estimated attending time of the new hall call. The system will allocate the new hall call to the elevator with the lowest total cost. That is,

$$e_{\text{candidate}} = \arg\min_{i \in [1,M]} t_i^{total} \tag{1b}$$

Lemma 1. The allocation based on (1b) is the best allocation under the immediate allocation policy.

Proof

Let $W_i^-, i \in [1, \mathrm{M}]$ denote the total waiting time of all the unattended passengers currently assigned to elevator *i*, excluding the new hall call.

Let $W_i^+, i \in [1, \mathrm{M}]$ denote the total waiting time of all the unattended passengers currently assigned to elevator *i*, including the new hall call.

Let $W_i\ , i \in [1, \mathrm{M}]$ denote the total waiting time of all the unattended passengers including the new hall call with the assigning the new hall call to elevator *i*,

$$\begin{aligned} W_i &= W_i^+ + \sum_{\substack{j=1 \\ j \neq i}}^{M} W_j^- \\ &= (W_i^+ - W_i^-) + \sum_{j=1}^{M} W_j^- \ . \end{aligned} \tag{2}$$

In formula (2), $\sum_{j=1}^{M} W_j^-$ is the total waiting time of all of unattended passengers before the new hall call arrives and it cannot be changed based on the immediate allocation policy. We divide two sides of formula (2) simultaneously by all of the unattended passengers including the new arrival passenger (this number can be viewed as fixed if we do not consider the future arrival of the passengers) and obtain the average waiting time of these passengers. Thus, minimizing average waiting time of all of the passengers is equivalent to minimizing W_{i} (the total waiting time of all the passengers), which in turn is equivalent to minimizing $(W_i^+ - W_i^-) = t_i^{total}$ □

4.2 Three-passage concept

The three-passage concept is illustrated in Fig. 1. It describes the relationship between the assigned hall calls and elevator motion status characterized by elevator direction and position. The elevator serves the hall calls sequentially as it reaches them. Passage one (P1) hall calls are calls supposed to be served without changing the elevator direction, passage two (P2) after the changing the direction once, and passage three (P3) hall calls after a second change of direction.

If the elevator is idle (direction is NONE), the direction of the elevator is first set and then the hall call passage can be determined based on above description. If the new hall call occurs at the same floor as the current floor of the elevator, the direction of the elevator becomes the same as that of the hall call. If the hall call occurs at the floor above (below) the current floor of the elevator, then the direction of the elevator becomes UP (DOWN).

To simplify the presentation, we only derive the relationship when the elevator is moving up. The formulas are symmetric when the elevator is moving down.

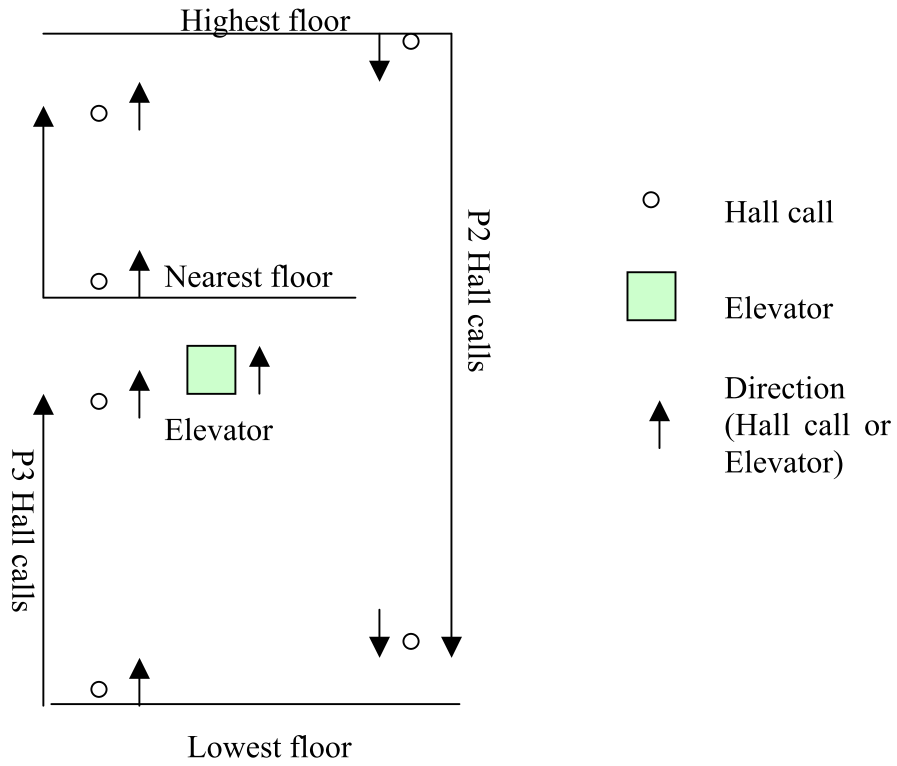

Fig. 1. Three Passage Concept.

The following notations are used.

M Number of elevators in the system

N Number of floors with the lobby floor indexed 0 and the top floor indexed by $N-1$.

$f_i^{nearest}$ The nearest floor of the i^{th} elevator.

d_i The direction of the i^{th} elevator, either UP or DOWN. If d_i is NONE (the elevator is not in motion in this case), d_i can be set first based on above description.

d The direction of the new hall call, either UP or DOWN.

k The floor of the new hall call.

H_i Set of hall calls assigned to the i^{th} elevator.

$H_i^{(1)}$ Set of P1 hall calls assigned to the i^{th} elevator.

$H_i^{(2)}$ Set of P2 hall calls assigned to the i^{th} elevator.

$H_i^{(3)}$ Set of P3 hall calls assigned to the i^{th} elevator.

We have

$$H_i^{(1)} = \left\{ < k, d > \in H_i \mid d = d_i \text{ and } k \geq f_i^{nearest} \right\} \tag{3}$$

$$H_i^{(2)} = \left\{ < k, d > \in H_i \mid d \neq d_i \right\} \tag{4}$$

$$H_i^{(3)} = \left\{ < k, d > \in H_i \mid d = d_i \text{ and } k < f_i^{nearest} \right\} \tag{5}$$

It is obvious that $H_i = H_i^{(1)} \cup H_i^{(2)} \cup H_i^{(3)}$, that is, the three-passage hall calls cover all the hall calls assigned to the elevator. Moreover, it is easy to extend the three-passage concept to describe the relationship between any hall call and any elevator in the system.

In the following, we will concentrate on implementation of the ETA estimation.

4.3 Implementation of ETA estimation

Formula (1a) provides the general guideline on how to evaluate the total cost of allocating the hall call to the elevator. The concrete implementation varies. For example, Nikovski & Brand (2003a) employed dynamic programming to estimate the total cost based on the assumption that the number of people that get off at the ongoing floors follow the binominal distribution. They did not explicitly include the stop cost.

Here we introduce stop cost explicitly. t_i^{total} is estimated based on the passage relationship between hall calls and the elevator *i*, the expected number of extra stops caused by the hall calls besides the mandatory stop at the occurred floor and the estimated elevator reversal floor. Before we describe ETA calculation, we briefly discuss the components of stop cost, the method of estimating the expected number of stops and estimated farthest floor of the hall call.

4.3.1 Expected extra stops and expected farthest floor

Stop cost is associated with the time delay caused by the elevator in serving the hall calls or car calls at the occurred floor. It consists of the following components: (a) open-door time, (b) close-door time, (c) photocell time (the time delay between the moment when the last passenger enters the elevator and the moment elevator starts closing the door), (d) entrance time (the time required for one passenger to enter the elevator), (e) exit time (the time required for one passenger to exit the elevator) and (f) acceleration/deceleration time (time lost due to acceleration and deceleration).

The impact of hall calls and car calls on elevator dynamics is different. Each car call only causes one mandatory stop but each hall call can cause a number of extra stops on the ongoing floors besides the mandatory stop at the occurred floor. The actual number of extra stops caused by the hall call varies based on the current assignment of the elevator including the car calls and assigned hall calls. In pure up-peak traffic the situation is greatly simplified. Hall calls will arrive only at the lobby floor, and the elevators attending them will be empty.

Next, we describe calculation of the expected number of extra stops and the expected farthest floor that the current hall call can reach. We extend the method by Siikonen (1997b) for dealing with the up-peak traffic pattern to any traffic situation. The following notations are introduced.

d	The direction of the new hall call, either UP or DOWN.
k	The floor of the new hall call.
n_k^{pass}	Number of passengers behind the hall call.
f_i^k	Number of floors to move to reach the farthest floor in the current direction.
l_i^{net}	Expected net distance to the farthest floor.
s_i^k	Expected number of stops for ongoing f_i^k floors caused by the new hall call <*k*,*d*> under the condition that the assignment of elevator *i* is empty.
$s_{k,j}^{extra}$	Actual extra number of stops caused by the new hall call <*k*,*d*> before it arrives at the j^{th} floor.
$s_{k,j}^{mandatory}$	Number of mandatory stops between floor *k* and floor *j* (excluding floor *k* and floor *j*).
$C_{k,j}$	Set of car calls between floor *k* and floor *j* (excluding floor *k* and floor *j*).
$H_{k,j}$	Set of assigned hall calls between floor *k* and *j* (excluding floor *k* and floor *j*) that should be attended before the elevator arrives at floor *j*.
$f_i^{farthest}$	Expected farthest floor that the current hall call can reach.
P_i^k	Probability that none of the passengers that board on floor *k* with travel direction *d* get off at each ongoing floor

To make the estimation simple, we assume that the passengers boarding on floor *k* have an equal probability to get off at the ongoing f_i^k floors. We have

$$P_i^k = \begin{cases} 1 - 1/f_i^k & \text{if } n_k^{pass} = 1 \\ e^{-n_k^{pass}/f_i^k} & \text{if } n_k^{pass} > 1 \end{cases}. \tag{6}$$

Thus,

$$s_i^k = f_i^k (1 - P_i^k) \tag{7}$$

and

$$l_i^{net} = \begin{cases} 0 & \text{if } f_i^k = 1 \\ \sum_{l=2}^{f_i^k} \prod_{j=f_i^k-l+1}^{f_i^k} P_i^k & \text{if } f_i^k > 1 \end{cases}. \tag{8}$$

After l_i^{net} has been calculated, it is trivial to obtain the expected farthest floor.

$$f_i^{farthest} = \text{the actual farthest floor - } l_i^{net} \tag{9}$$

The number of mandatory stops and extra stops are then estimated by

$$s_{k,j}^{madatory} = | C_{k,j} | + | H_{k,j} | - | C_{k,j} \cap H_{k,j} | \tag{10}$$

and

$$s_{k,j}^{extra} = s_i^k (| j - k | - 1 - s_{k,j}^{mandatory}) / f_i^k . \tag{11}$$

4.3.2 Total time t_i^{total} estimation

(1) Attending time $t_i^{attending}$ estimation

t_s Stop time of one full stop. It is the sum of 6 components described at the beginning of this subsection.

C_i Set of car calls currently in the i^{th} elevator.

H_i Set of hall calls assigned to the i^{th} elevator.

C_i^{before} Set of car calls to be attended before the hall call <*k*,*d*>.

C_i^{after} Set of car calls to be attended after the hall call <*k*,*d*>.

H_i^{before} Set of hall calls to be attended before the hall call <*k*,*d*>.

H_i^{after} Set of hall calls to be attended after the hall call <*k*,*d*>.

$t_{i,k}^{nonstop}$ Nonstop travel time to floor *k* based on current motion status of the elevator *i*.

Then $t_i^{attending}$ is estimated as follows.

(a) If <*k*,*d*> is a P1 hall call, then

$$t_i^{attending} = t_{i,k}^{nonstop} + (| C_i^{before} | + | H_i^{before} | - | C_i^{before} \cap H_i^{before} | + \sum_{i \in H_i^{before}} s_{i,k}^{extra}) t_s \tag{12}$$

(b) If <*k*,*d*> is a P2 hall call, then

(i) If no P1 hall calls and car calls exist in elevator *i* , $t_i^{attending}$ can be estimated using the same logic as (12)

(ii) Otherwise, $t_i^{attending}$ can be decomposed into two parts: The first part is the time to finish serving all of car calls and P1 hall calls if they exist, then the elevator reverses the direction. The second part is the time to attend <*k*,*d*> is a from the reversal floor. The reversal floor can be calculated by combination of the estimated farthest floors of $H_i^{(1)}$ hall calls based on (9) and C_i (set of car calls currently in elevator *i*). Two parts of time can be estimated separately using the same logic as in (12).

(c) If <*k*,*d*> is a P3 hall call, then $t_i^{attending}$ can be decomposed into three parts: The first part is the time to finish serving all of car calls and P1 hall calls if they exist, then the elevator reverses the direction. The second part is the time to finish serving all the P2 hall calls if they exist, then the elevator reverses the direction for the second time, the last part is the time to attend <*k*,*d*> from the second reversal floor. The first reversal floor can be calculated by combination of the estimated farthest floors of $H_i^{(1)}$ hall calls based on (9) and C_i and the second reversal floor can be determined similarly based on the farthest floors of $H_i^{(2)}$. Three parts of time can be estimated separately using the same logic as in (12). In some cases, either the first part or second part time does not exist, but the estimation logic remains unchanged.

(2) SDF t_i^{delay} estimation.

The delay time $t_{i,j}^{delay}$ that the new hall call <*k*,*d*> will cause to passenger *j* associated with H_i^{after} is made up of three parts:

(a) One mandatory stop at the occurred floor of the evaluated hall call if there is no committed car call at the very floor. In other words, if the floor of the hall call coincides with the car call in the right direction, this stop time can be ignored because the elevator has to stop at this floor for letting off the passenger(s) regardless of the occurrence of the hall call;

(b) The expected number of extra stops on the ongoing floors caused by the hall call and

(c) Additional travel time if the hall call will affect the reversal floor.

For regular SDF estimation

$$t_i^{delay} = \sum_{j \in H_i^{after}} t_{i,j}^{delay} n_j^{pass} \tag{13}$$

where n_j^{pass} is the number of passengers behind the hall call.

4.4 Basic ETA allocation algorithm

In view of the fact that the queue length information behind the hall call is unknown in the real elevator system, we take the number of passengers behind the hall call as one when we implement the algorithms. The estimation based on Section 4.3 is still valid.

The procedures for the basic ETA allocation algorithm are summarized as follows

Step 1. For all of the elevators

Step 1.1 Determine the passage of the new hall call <*k*,*d*> with respect to each elevator $i \in [1,M]$.

Step 1.2 Calculate the total cost t_i^{total} that is the cost of assigning the <*k*,*d*> hall call to elevator *i* based on the formula described in Section 4.3 depending on the passage of <*k*,*d*>.

Step 2. Choose the best allocation for the new hall call <*k*,*d*> based on (1b)

The time complexity of the basic ETA allocation algorithm is O(MN).

5. Reallocation variant ETA allocation algorithm

The basic ETA allocation algorithm employs the immediate allocation policy. Some early-allocated hall call may lose optimality as future events appear in the system. Our reallocation variant of the ETA algorithm is based on the coordination between the basic ETA algorithm and a heuristic reallocation mechanism to improve the performance of the allocation algorithm further.

5.1 Candidates for hall call reallocation

Three categories of hall calls will be considered for reallocation.

(1) The oldest call in the system.

(2) Some hall calls of the elevator that is about to leave the current floor:

(a) The oldest hall call if the number of hall calls assigned to the elevator exceeds a certain limit.

(b) The hall call whose attending time can be severely affected. This category includes the last hall call in the service sequence if it belongs to the P2 or P3 passage.

(c) The hall calls whose attending time is relatively certain, for example, the successive hall call and the hall calls that coincide with the car calls in the right direction (called matched hall calls).

5.2 The reallocation mechanism

To reduce the computational effort, the candidates for reallocation hall calls are reevaluated serially. Before reallocation, all of the hall calls are assigned to the elevator based on the basic ETA allocation algorithm. Each time only one hall call is considered for reallocation, but a finite number of hall calls may be reevaluated in sequence for one reallocation slot. The following time slots are chosen as the reallocation points.

(1) Every time when a new hall call arrives, the basic ETA allocation algorithm finds the appropriate elevator for the new hall call and then the oldest call in the system (if it exists) is considered for reallocation.

(2) When the elevator is about to leave the current floor. First, the oldest hall call in the system is considered for reallocation. Then, the hall calls of this elevator (introduced in

section 5.1) are considered for reassignment. If there is overlap among these hall calls, reallocation is considered just once for the same hall call.

5.3 Coordination between different categories of reallocation hall calls

It is necessary to coordinate between different categories of hall calls so that their aggregate effect moves toward improving the performance of algorithm. Timing is important. We introduce threshold in reallocation. That is, the reallocation is executed only when the hall calls stay in the system for a sufficient period of time in terms of the oldest hall call, the last hall call to be served and the oldest call of the elevator. There is no such restriction on the matched hall calls.

5.4 Reallocation variant ETA allocation algorithm

The reallocation variant of the ETA algorithm can be viewed as interaction between triggering hall calls and the basic ETA allocation algorithm in section 4.4 at the appropriate moment. The reallocation ETA group control algorithm framework is illustrated in Fig. 2. Obviously, the time complexity of the reallocation ETA allocation algorithm is still O(MN). Introducing the reallocation mechanism based on section 5 does not affect the time complexity because the number of reallocation hall calls for each relocation slot is finite. The practical computation time should be k-fold ($k \geq 1$) of the computation time of the basic ETA allocation algorithm depending on how many hall calls enter the reallocation for each reallocation slot.

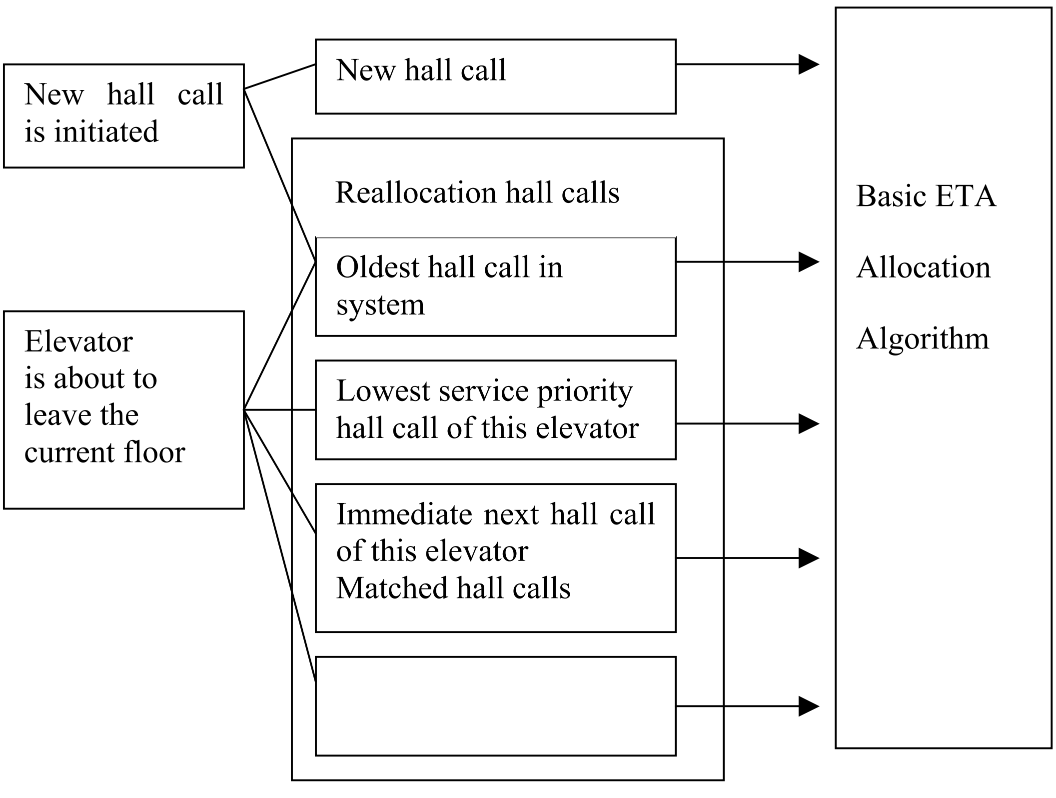

Fig. 2 Reallocation variant ETA Group Control Algorithm Framework

6. Test runs

Our basic ETA allocation algorithm (without reallocation) and its reallocation variant were tested through simulation. To verify the estimation accuracy of the basic ETA allocation algorithm, we benchmarked it against the commercially available ETA allocation algorithm on *Elevate* developed by Peters Research Ltd. (2002). The reallocation variant ETA was compared with the basic ETA allocation algorithm to verify the effect of the reallocation mechanism. Three performance criteria were used in the comparisons: average waiting time (AWT), average journey time (AJT) and the percentage of passengers that wait longer than 60 seconds (P > 60s).

- AWT: The average waiting time of all the passengers. Passenger waiting time is defined as the actual time a prospective waits after registering a landing call (or entering the waiting queue if a call has already been registered) until the passenger has entered the elevator.
- AJT: The average journey time of all the passengers. Passenger journey time begins when the waiting time begins and ends when passenger has exited the elevator.

Elevate defines the waiting time as starting from the passenger arrival and ending at the elevator arrival at the origin floor. Journey time is from the passenger arrival to the elevator arrival at the target floor. Therefore the events of each passenger in *Elevate* were reconstructed to compute the waiting times and journey times according to our definitions.

The benchmark ETA allocation algorithm is the group control algorithm without reallocation. The following notations are used to distinguish between the different variants of ETA allocation algorithms.

ETA-E: Benchmark ETA algorithm on *Elevate*

ETA-U: Basic ETA algorithm without reallocation in this paper on our simulator

ETA-R: Reallocation variant ETA algorithm in this paper on our simulator

6.1 Test problem

To be representative for a wide range of practical situations, we have generated the test problems based on the actual buildings by considering building types, elevator configurations and traffic scenarios. Table 1 shows the building parameters and related elevator configurations. The number of floors ranges from 9 to 40 and the number of elevators from 3 to 8. Building B has two entrance floors. The elevator group of building C serves the upper zone and elevators do not stop at floors 1-24. The bottom floor is indexed 0.

Passenger traffic in the building can be described as combinations of the three basic components: Incoming, outgoing and inter-floor traffic.

- Incoming passenger arrives at entrance floor and travels up to the populated floor in the building.
- Outgoing passenger arrives at populated floor and travels down to the entrance floor.

- Inter-floor passenger arrives at populated floor and travels to another populated floor.

Four traffic patterns used in the simulation test are shown in Table 2.

For each traffic pattern, traffic density was generated at two intensity levels: full and half arrival rates respectively as shown in Table1. Arrival rate is defined as percentage of population in five minutes. Therefore, a total of eight traffic scenarios were investigated: Heavy incoming (HI), moderate incoming (MI), heavy outgoing (HO), moderate outgoing (MO), heavy lunch (HL), moderate lunch (ML), heavy two-way (HT) and moderate two-way (MT). For each traffic scenario, ten random samples of one-hour traffic were generated. Each building contains 80 test problems and altogether 240 test problems were generated for three buildings.

In practice, the performance in pure up-peak situation can be improved greatly if elevators are returned to the entrance after they finish their current commitments even if no calls are allocated to them. There are several possible methods to return elevators. Here we concentrate on the performance of allocation algorithms and simulators are not equipped with returner algorithms. Instead, the incoming traffic pattern contains 5 % outgoing passengers, which return elevators to the entrance floor(s).

Table 1. Building parameters and the related elevator configurations.

Parameter	Building A	Building B	Building C
Total floors	9	16	40
Elevators	3	4	8
Capacity (persons)	13	20	21
Door opening time (s)	1.9	1.2	1.6
Door closing time (s)	2.8	2.5	2.6
Door pre-opening (s)	0	0	0
Photocell delay (s)	0.9	0.9	0.9
Loading time (s)	1.2	1.0	1.0
Unloading time (s)	1.2	1.0	1.0
Max velocity (m/s)	1.0	2.5	4.0
Acceleration (m/s^2)	0.8	1.0	1.0
Start Delay (s)	0	0	0
Floor height (m)	3.8	3.6	3.6
Exception heights (m)	Floor 0: 4.6 m	Floor 0: 3.9 m	Floor 0: 4 m
Lobby floor (entrance %)	Floor 0 (100%)	Floor 0 (20%) Floor 1 (80%)	Floor 0 (100%)
Population by floors	Floors 1,7,8: 30, other floors: 70	Floors 2-9: 20, floors 10-16: 16, floor 15: 2	Floors 25-39: 90
Total population	440	242	1350
Full arrival % of population / 5 min	15	40	13
Half arrival % of population / 5 min	7.5	20	6.5

Table 2 Traffic patterns

Traffic pattern	Incoming (%)	Outgoing (%)	Inter-floor (%)
Incoming	95	5	0
Outgoing	0	100	0
Lunch	40	40	20
Two-way	50	50	0

6.2 Test results

Table 3. Performance comparison for different variants of ETA allocation algorithms.

Build.	Traffic	ETA-E			ETA-U			ETA-R		
		AWT	AJT	P>60s	AWT	AJT	P>60s	AWT	AJT	P>60s
	HI	26.0	79.9	5.68	22.6	75.8	2.86	23.0	76.6	3.51
	MI	16.2	58.7	0.25	14.9	57.3	0.05	14.9	57.4	0.03
	HO	37.4	78.1	20.52	29.3	70.6	10.21	26.3	69.7	6.40
A	MO	27.7	60.3	5.93	21.3	54.4	1.69	21.1	54.5	1.16
	HL	38.4	79.4	23.25	34.6	75.3	18.09	31.6	73.6	14.51
	ML	21.0	51.0	3.13	18.5	48.5	2.90	18.3	48.4	1.41
	HT	34.5	76.5	18.41	30.8	72.3	13.62	29.4	71.9	12.69
	MT	19.8	51.3	2.22	17.0	48.4	2.45	16.0	47.6	0.91
	AVG	27.6	66.9	9.92	23.6	62.8	6.49	22.6	62.5	5.08
	HI	19.0	76.0	1.79	16.4	72.7	1.27	15.9	72.5	0.67
	MI	13.0	49.8	0.00	11.2	47.1	0.05	11.2	47.5	0.00
	HO	31.5	69.9	11.91	22.5	62.6	3.52	21.3	61.8	1.70
B	MO	21.0	47.2	1.43	15.2	41.6	0.17	15.1	41.8	0.05
	HL	33.4	76.0	16.42	31.1	71.7	13.11	27.8	69.9	8.99
	ML	15.6	39.4	0.62	12.9	36.6	0.53	12.3	36.3	0.07
	HT	28.9	71.0	11.05	26.2	66.0	8.66	23.9	65.0	4.96
	MT	13.6	37.5	0.15	10.9	35.0	0.38	10.6	34.7	0.02
	AVG	22.0	58.4	5.42	18.3	54.2	3.46	17.3	53.7	2.06
	HI	33.6	131.0	15.20	30.4	127.9	13.53	31.6	129.0	15.33
	MI	17.0	104.5	0.14	14.9	104.5	0.02	14.9	104.5	0.01
	HO	34.5	101.7	16.83	33.1	100.7	14.08	26.9	96.7	7.96
C	MO	33.2	88.7	14.93	22.9	85.3	1.71	22.2	85.2	0.81
	HL	35.3	107.8	20.58	27.3	101.4	10.53	24.8	99.9	8.31
	ML	20.9	71.3	4.02	15.2	66.0	1.21	14.5	65.7	0.37
	HT	33.5	108.5	17.78	28.6	104.4	10.81	22.6	100.7	6.93
	MT	19.2	74.2	3.16	14.1	69.4	1.28	13.3	69.2	0.35
	AVG	28.4	98.5	11.58	23.3	94.9	6.65	21.3	93.9	5.01

Table 3 shows the results for different variants of ETA allocation algorithms against 8 traffic scenarios. Based on the results, we make the following observations.

(1) Compared with ETA-E, the reduction in AWT for ETA-U is in the range [4%, 31%] with an average of 16 %. For AJT, the reduction is in the range [0.1%, 12%] with an average of 5 %.

(2) In terms of long waiting percentage (P>60), the improvement for ETA-U is in the range [-0.3%, 13%] with an average of 3 % points.

There is a phenomenon that ETA-U improves the average waiting time of moderate down-peak traffic scenario significantly as compared with the improvement of heavy down-peak traffic scenario. The reason behind it is that the performance of ETA-U is limited by the elevator capacity. We did not consider the capacity when the assignment was done. The elevator passes the landing hall call when the load is equal to or greater than the bypass load. Reallocation is a remedy.

(3) As compared with ETA-U, the reallocation variant ETA-R can improve the performance consistently for all the traffic scenarios except the up-peak traffic scenario. Its overall speed-up of AWT is 7 % and the reduction of long waiting percentage is 2 % points. Generally, reallocation can improve the performance of the heavy traffic scenario more significantly. For heavy traffic patterns, it can reduce AWT by more than 10 % and the long waiting percentage by more than 3 % points.

The reason why reallocation worsens the performance for the up-peak traffic scenario may lie in the fact that the elevators compete for one hall call and paralyze the reallocation mechanism. On the way back to the lobby floor, the elevator may stop halfway on the intermediate floor because the reallocation can transfer the lobby floor hall call from elevator to elevator and thus delay the transportation of the passengers on the lobby floor. From Table 3, we can also see that the reallocation can improve the performance in the heavy up-peak traffic scenario for building B. This is not occasional because building B has two entrances that can generate hall calls in up-peak traffic scenarios, which can make the elevator return the lobby floors more securely and transport the passengers on the lobby floors timely. Therefore, the strength of the reallocation lies in its ability to coordinate the allocation among different hall calls instead of competing for the same hall call.

(4) In terms of AWT, the effect of reallocation is not significant. The improvement is in the range [-1%, 4%] with an average of 1 %.

7. Conclusions

We have developed ETA based elevator group control algorithms with more accurate estimation. A heuristic reallocation mechanism improves the performance further. In the test runs with real-life building types, elevator configurations and typical traffic scenarios, our basic ETA-U allocation algorithm (without reallocation) (as compared with the benchmark ETA) reduced the waiting time by 16 % and the percentage of the passengers who wait for more than 60 seconds by 3 % points. In addition, ETA-U can bring 5 % improvement on the journey time. The reallocation variant ETA-R can enhance the performance further as compared with ETA-U. The further improvement in waiting time is 7 % and the reduction in the long waiting percentage is 2 % points.

References

Chan W.L., So A.T.P., Lam K.C., 1997. Dynamic zoning in elevator traffic control. Elevator world, March, 136-140.

Cho Y.C., Gagov Z., Kwon W.H., 1999. Timed Petri net based approach for elevator group controls. Technical report No. SNU-EE-TR-1999-3, School of electrical engineering, Seoul National University.

Crites R.H., Barto A.G., 1996. Improving elevator performance using reinforcement learning. In D.S. Touretzky, M.C. Mozer, and M.E. Hasselmo (eds.) Advances in Neural Information Processing Systems 8, MIT Press, Cambridge MA, 1017-1023.

Fujino A., Tobita T., Segawa K., Yoneda K., Togawa A., 1997. An Elevator group control system with floor-attribute control method and system optimization using genetic algorithms. IEEE Transactions on Industrial Electronics, 44(4), 546-552.

Gagov Z., Cho Y.C., Kwon W.H., 2001. Improved concept for derivation of velocity profiles for elevator systems. Proceedings of the 2001 IEEE International Conference on Robotics & Automation, Seoul, Korea, May 21-26, 2419-2423.

Gamse B., Newell G. F., 1982a. An analysis of elevator operations in moderate high buildings-I - Single elevator. Transportation research-B 16(4), 303-319.

Gamse B. & Newell G. F., 1982b. An analysis of Elevator Operations in moderate high buildings-II - Multiple elevators. Transportation research-B 16(4), 321-335.

Gandhi A.D., Cassandras C.G., 1996. Optimal control of polling models for transportation applications. Math. Comput. Modeling 23(11/12), 1-23.

Hakonen H., 2003. Simulation of building traffic and evacuation by elevators. Licentiate thesis, System Analysis Laboratory, Helsinki University of Technology.

Kim C.B., Seong K.A, Hyung L.K., Kim J.O., 1998. Design and implementation of a fuzzy elevator group control system. IEEE Transactions on systems, Man and Cybernetics—Part A: Systems and Humans. 28(3), 277-287.

Nikovski D., Brand M., 2003a. Decision-theoretic group elevator scheduling. 13th International Conference on Automated Planning and Scheduling, ICAPS'03, June 9-13, Trento, Italy.

Nikovski D., Brand M., 2003b. Marginalizing out future passengers in group elevator control. 9th Conference on Uncertainty in Artificial Intelligence, UAI-2003, August 8-10, Acapulco, Mexico.

Pepyne D.L., Cassandras C.G., 1998. Design and implementation of an adaptive dispatching controller for elevator systems during up-peak Traffic. IEEE Transactions on Control Systems Technology, 6(5), 635-650.

Peters Research Ltd., 2002. Getting started with Elevate.

Siikonen M.L., 1997a. Elevator group control with artificial intelligence. Research reports A 67, System Analysis Laboratory, Helsinki University of Technology.

Siikonen M.L., 1997b. Customer service in an elevator system during up-peak. Transportation research-B, 31(2), 127-139.

Siikonen M.L., Susi T., Hakonen H., 2001. Passenger traffic flow simulation in tall buildings. Elevator world, 117-123.

Smith R., Peters R., 2002. ETD algorithm with destination dispatch and booster options. Elevator world, July, 136-142.

Sutton R.S. , Barto A.G., 1998. Reinforcement learning: An introduction. MIT press.

Tyni T., Ylinen J., 1999. Improving the performance of Genetic Algorithms with a Gene Bank. Proceedings of EUROGEN99, Report A2/1999, University of Jyväskylä, Finland, 162-170.

Turku Centre for Computer Science
Lemminkäisenkatu 14
FIN-20520 Turku
Finland

http://www.tucs.fi/

University of Turku
• Department of Information Technology
• Department of Mathematics

Åbo Akademi University
• Department of Computer Science
• Institute for Advanced Management Systems Research

Turku School of Economics and Business Administration
• Institute of Information Systems Science