Landing Call Allocation Based on Linear Programme Method

Chen Guifeng
Chevalier Singapore Holdings Limited

Keywords: Linear Programme, Minimum Cost, Waiting Time, Dispatching, Hungarian Method.

Abstract:

This paper describes a way of landing call allocation, which is based on the linear programme algorithm in elevator group control system. The basic idea here is to minimize the total cost – the waiting time of passengers and so enhance the elevator system's performance. The paper discusses the simulation of the waiting time based on the given system using general concept and theorem to construct the idea of the linear program, which is a classical optimizing theorem. The example demonstrated how the linear programme works and the application in the landing call allocation.

1. Introduction

People, even from elevator industry, wander how the elevator responds their calls. Most of them thought the elevator with shortest responding time will answer their call. The actual answer for this is different. It depends on the condition you are in. For the simplex system, the calls are answered one by one at the same direction up to the last, then reverses. The response time or the passenger waiting time varies greatly because of the unforeseen events at each landing (1). But for multiple elevator system, there exists the call allocation problem, the system must decide the proper car to answer related call. For example, a 5 elevator system with group control and intelligent car dispatching system, when a call is made, it will firstly find out the position of the call, time period, e.g., up/down peak hour, morning time, afternoon or others. How many other calls are waiting and how long of each call has been waiting for already. Based on these informations, the system then can determine the most suitable elevator with minimum cost to response the call according to the certain optimised algorithm used in this system.

2. Landing call allocation procedure (2)

The task of the landing call allocation procedure is to allocate a suitable car to each landing call. The existence of a landing call queue simplifies this procedure as the queue defines the order in which the allocation procedure considers the landing calls. A landing call is taken from the head end of the queue and allocated to a suitable car using the concept of minimum cost. This concept operates by performing a trial allocation to all available cars and allocating the call to the car presenting the lowest cost.

As a conventional two button signalling system is being considered, this means only passenger waiting time can be minimised. The period of duration for a particular landing call only represents the time the first passenger at that landing has to wait. All subsequent passengers benefit from the first registration and actually wait for less time. Thus the control algorithm can only reduce the cost of the system response time to service a landing call. Fig 2.1 indicates that AWT (Average Waiting Time) is very nearly equal to ASRT (Average System Response Time) for balanced inter-floor traffic so for this traffic condition, this cost function is very suitable. (Here, we only concern the condition under balanced inter-floor traffic). For the up peak , the ASRT is related to interval INT by values of car load and for down-peak traffic the AWT is linearly related to demand. Special measures must be taken to deal with unbalanced traffic conditions.

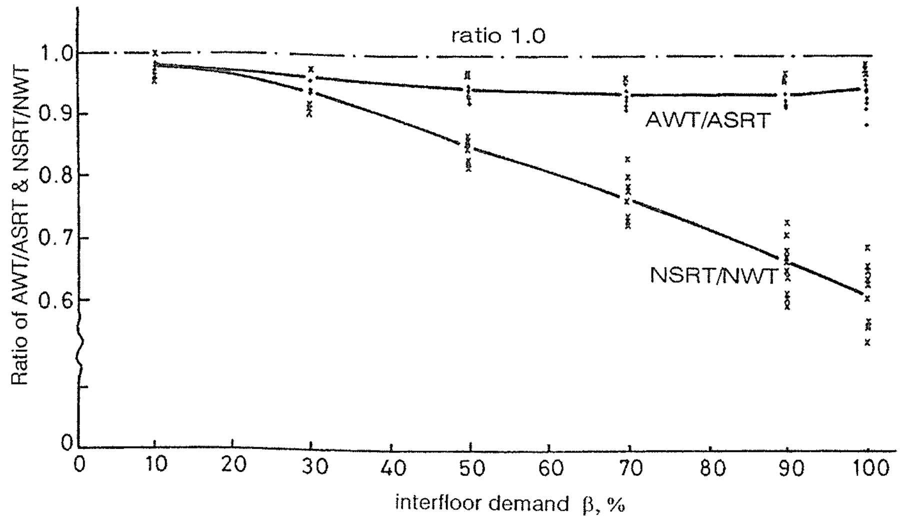


Fig. 2.1 Ratios of average waiting time (AWT) to average system response time (ASRT) and number of landing calls (NSRT) and number of passengers (NWT) for interfloor traffic

The cost of an allocation of a new landing call to a particular car is given by the incremental value of the expected extra system response time due to the new allocation. Applying this procedure to evaluating the cost to allocate a new call to each car in turn will result in the minimum cost allocation being selected. In the special case, when all cars are not committed to any allocations and a new landing call occurs, then the cost will be represented by the system response time of the nearest car to the call. The allocation procedure thus has an in-built capability of assigning a landing call to a nearest car.

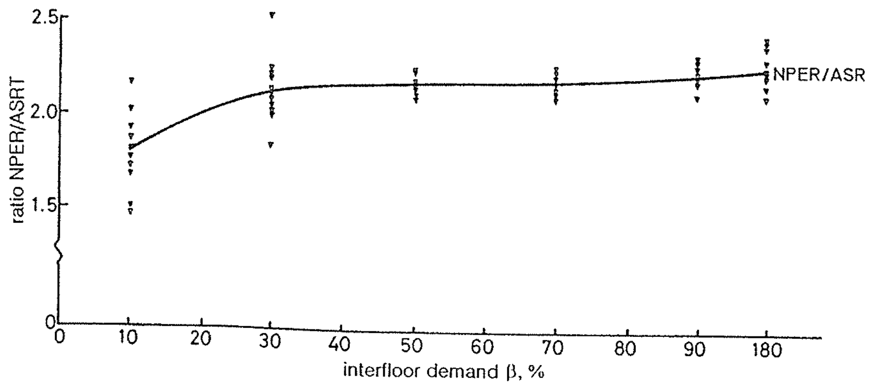


Fig. 2.2 Ratio ninety percentile (NPER) to average system response time (ASRT) for interfloor traffic

In order to apply the minimisation procedure it is necessary to calculate the car journey times to travel to a landing call. These journey times may be on the basis of a direct trip to a landing or an indirect trip where the car stops an intermediate landing on its journey. A car journey once assigned to a call then becomes the calculated (estimated) system response time for that call. A car journey time consists of several components:

1) Inter-floor flight time (including acceleration, deceleration levelling and travel at contract speed).
2) Door operating time (opening and closing);

3) Passenger transfer time.

There are difficulties in calculating car journey times. Each time a car stops it is not known how many passengers will enter or leave the car. This has the effect of making item 3) (passenger transfer time) difficult to evaluate.

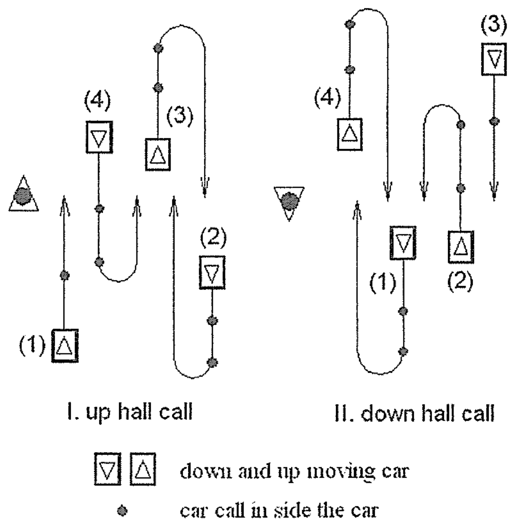
Fig. 2.1 indicates 90% of the time only one passenger enters a car at a landing call floor. Fig. 2.2 shows 90% of the time, only one passenger gets out at each stop (Refer to Page. 319, [2]).

The linear programming here is a dynamic algorithm because the elevator system itself is a dynamic. So it's not a simple problem. The linear programme itself is the way used in the task dispatch or the transport assignment in order to make the things more efficiency and most economic. It's normally used in the market strategy and analysis, transportation assignment and so on.

3. The Simulation of Response Time or car journey time

[2] gives more detail information to describe the selection of door operation time, the passenger boarding and alighting times. And also the number of passengers in the car, though it is a very hard factor to estimate, can be determined based on the above mentioned derivation.

The interfloor flight can have many types, which are shown as figure right. When the car is standstill, it's easy to determine, the figure only shows the moving car with car calls inside. In each up/down hall call, there have 4 cases.



So we can define the interfloor flight time as *IFFT*:

$$IFFT=(n+1)\times(T_u+T_d)+T_c+P\times T_p+(n+1)\times T_d$$

T_u: acceleration time (speed up time)
T_d: deceleration time (slowdown time)
T_c: time under contract speed.
n: number of stops between the car and the destination floor,
P: the estimated passenger number to transfer
T_p: passenger transfer time
T_d: door operation time (opening and closing door).

The n, P, Tp and Td can be found according to the proposed method stated as above.

Theoretically, the T_u and T_d can be taken as an equivalent constant value. Although, they may be slightly different, it can be ignored. T_u can be calculated from acceleration, jerk and the contract speed. For example,

$$V=v_0+a\times t, \text{ so } T_u=V_c/a$$

T_c is the duration of the elevator running under the contract speed (constant), what we need is to compute the distance running under this speed. The floor height is fixed, take away the distances in the speeding up and slowing down period, it is the actual distance under contract speed. For the speed up/ slowdown period, the distance is

$$S = 2\int_0^{Tv} vdt = at^2\Big|_0^{Tv} = aT_v^{\,2}$$

So, $T_v = \dfrac{H - aT_v^{\,2}}{Vc}$ V_c is the contract speed in m/s.

That is *IFFT*.

4. The Linear Programming: The hall call allocation

The landing call allocation is a special type of linear programming problem. For a 22 storeys, 5 lifts system, the lifts are arranged in a line. If, at 10^{th} floor, up call is activated, which one of the 5 lifts is assigned to respond it? When all lifts are standstill, the problem becomes very simple, if all the lifts are in different storeys and running in different directions, there also have several car calls in side. Then how? The problem becomes difficult. Here, the assignment method of linear programming is introduced to solve this problem.

4.1 The Assignment Concept

Given n lifts to be assigned to n calls, with c_{ij}, the cost of assigning lift i to call j, that means, the responding time from present situation to answer the call, find an assignment to minimum total cost.

Following assumptions must be satisfied to fit the definition of an assignment problem, it need to be formulated in such a way:

1. The number of lifts and the number of calls are the same. (say n).
2. Each lift is to be assigned to exactly one call.
3. Each call is to be performed by exactly one lift.
4. There is a cost c_{ji} associated with lift i (i=1, 2, , , , , n) responding call j (j=1, 2, , , n).
5. Then, the objective is to determine how all n lifts should be made in order to minimize the cost.

Dummy lifts or dummy calls can be used to satisfy the first 3 assumptions, that means, if the call number is less than the lift number, the dummy calls must be added in order to keep the same number with lifts; if the call number is more than that of lifts, then the dummy lifts must be added to meet the condition 1.

4.2 The Model of Assignment Problem

The assignment problem uses the following decision variables:

X_{ij}=1 if lift i responses call j, Xij=0 if not.

By letting Z denote the total cost, the assignment problem model is:

$$\text{Minimize } Z = \sum_{i=1}^{n}\sum_{j=1}^{n} c_{ij}x_{ij}$$

Subject to: $\sum_{j=1}^{n} x_{ij} = 1 \;\; i=1,,,,n$

$\sum_{i=1}^{n} x_{ij} = 1 \;\; j=1,,,n;\;\; x_{ij} \geq 0$ for all i and j (x_{ij}, binary, for all i and j)

We take the above example. The system now has 5 lifts and 5 calls during certain time. According to the rules above, each lift must be assigned to response one call. The time required to set up each lift for completing each call is shown in the Table 1.

Table 1: Response time of each lift to each call (s)							
Lift Status			Call floor & Type				
Lift	Position	Direction	21D	16U	9U	8D	4D
L1	10	Up	30.34	15.34	14.89	17.09	25.89
L2	2	Down	54.49	43.49	28.09	25.89	17.09
L3	15	Down	25.89	14.89	15.34	17.54	26.34
L4	7	Up	36.94	21.94	6.54	8.34	19.29
L5	19	Up	10.54	19.29	34.69	36.89	45.69

Comments:

21D: 21 storey, Down Call activated; 16U: 16 storey, Up call activated; others are same defined.
The data in the table is based on the following assumption and simulated:
Contract speed: 1.5*m/s*, Acceleration & deceleration: 0.7m/s^2; Door opening / closing time: 4s; Passenger transfer time: 1.2*s*; Standard floor height: 3.3*m*; Number of stops: 22.

4.3 Set up times for lift system

The system wants to minimize the total setup time needed to response the 5 calls, we define (for i, j = 1, 2, 3, 4, 5)

X_{ij}=1 if lift ***i*** is assigned to meet the demands of call *j*; X_{ij}=0, if not.

Then the problem is formulated as:

$$\text{Min } Z = \sum_{i=1}^{5}\sum_{j=1}^{5} C_{ij} x_{ij}$$

subject to:

$$\text{elevator constraints} \begin{cases} \sum_{i=1}^{5} x_{1i} = 1 \\ \sum_{i=1}^{5} x_{2i} = 1 \\ \vdots \\ \sum_{i=1}^{5} x_{5i} = 1 \end{cases} \qquad \text{call constraints} \begin{cases} \sum_{i=1}^{5} x_{i1} = 1 \\ \sum_{i=1}^{5} x_{i2} = 1 \\ \vdots \\ \sum_{i=1}^{5} x_{i5} = 1 \end{cases}$$

with x_{ij}=0 or x_{ij}=1

If x_{ij}=1, the objective function will pick up the time required to set up lift *i* for call *j*; if x_{ij}=0, the objective function will not pick up the time required to set up lift *i* for call *j*.

For the above lift system, Table 1 lists the response times, i.e., x_{ij}=1. Because all lifts are possible to response the calls, otherwise x_{ij}=0.

The cost table including identifying the lifts and calls is shown below, this table contains all the essential data in a far more compact form.

Table 2. Cost Table				
	1	2	...	N
1	C_{11}	C_{12}	...	C_{1n}
2	C_{21}	C_{22}	...	C_{2n}
.	.	.		.
N	C_{n1}	C_{n2}	...	C_{nn}

4.4 Hungarian Method

***Hungarian* method** is the efficient way to solve assignment problem, it operates directly on the cost matrix and converts the original cost table into a series of equivalent cost tables until it reaches one where an optimal solution is obvious. This final equivalent cost table is the one consisting of only positive or zero elements where all the assignments can be made to the zero element positions. This set of assignments will be a zero. Then the total cost is optimal.

Four steps to solve these problems:

Step 1. In the cost table, locate the *smallest element* and subtract it from *every element in that row*. *Repeat* this procedure *for each column* (the column minimum is determined after the row subtractions). The revised cost matrix will have at least one zero in every row and column.

Step 2. Determine whether there exists a feasible assignment involving only zero costs in the revised cost matrix, i.e., to find if the revised matrix has n zero entries no two of which are in the same row or column. If exists, it is optimal. If not, go to step 3.

Step 3. Cover all zeros in the revised matrix with as few horizontal and vertical lines as possible. The total number of lines in this minimal covering will be less than n. Locate the smallest number from every element not covered by a line. Subtract this number from every element not covered by a line and add it to every element those lines at the intersection of two lines.

Step 4. Return to step 2.

4.5 Assignment Problem for the lift call allocation

For an assignment problem, the cost C_{ij} is the total cost associated with elevator i to response call j. We consider the cost table above, it's a exact $n \times n$ matrix, it need not to do any adjustment. So, there are $m=n=5$ assignment to be made. Use Hungarian method to solve this problem.

We find the smallest element in the table and subtract it from every element in that row. Then repeat this for each column, we get the table below.

Table 3. Response time of each lift to each call (s)							
Lift Status			Call floor & Type				
Lift No.	Position	Direction	21D	16U	9U	8D	4D
L1	10	Up	19.8	0.45	14.89	15.29	13.14
L2	2	Down	43.95	28.6	28.09	24.09	4.34
L3	15	Down	15.35	0	15.34	15.74	13.59
L4	7	Up	19.86	0.51	0	0	0
L5	19	Up	0	4.4	34.69	35.09	32.94

Only 2 assignments to 0-element position are possible. Cover all 0's. The minimum element not crossed out is 4.34 in the last column and L2 row. Subtract every element not covered by a line by 4.34. Then, we can find 3 assignments to 0-element position is possible, cover all 0's again, the minimum element not crossed out is 10.55 in the last 3rd column and L1 row. Subtract every element not covered by a line by 10.55. We then have following table.

Table 4. Response time of each lift to each call (s)							
Lift Status			Call floor & Type				
Lift	Position	Direction	21D	16U	9U	8D	4D
L1	10	Up	19.8	0.45	0	0.4	8.8
L2	2	Down	43.95	28.6	13.2	9.2	0
L3	15	Down	15.35	0	0.45	0.85	9.25
L4	7	Up	34.75	15.4	0	0	10.55
L5	19	Up	0	4.4	19.8	20.2	28.6

Cover all 0's. 5 assignments to 0 element position is possible. So, we can make the complete assignment shown as following table (element with ✓).

Table 5. Response time of each lift to each call (s)							
Lift Status			Call floor & Type				
Lift	Position	Direction	21D	16U	9U	8D	4D
L1	10	Up	19.8	0.45	0✓	0.4	8.8
L2	2	Down	43.95	28.6	13.2	9.2	0✓
L3	15	Down	15.35	0✓	0.45	0.85	9.25
L4	7	Up	34.75	15.4	0	0✓	10.55
L5	19	Up	0✓	4.4	19.8	20.2	28.6

The resulting total cost from Table A is:

$$Z^* = c_{13}+c_{25}+c_{32}+c_{44}+c_{51} = 14.89+17.09+14.89+8.34+10.54=65.75 \text{ (seconds)}$$

The optimal solution for this assignment problem is: Lift 1 to storey 9 up call; Lift 2 to storey 4 down call; Lift 3 to storey 16 up call; Lift 4 to storey 8 down call; Lift 5 to storey 21 down call.

The total cost is Z=65.75s, that is the total waiting time for these 5 call is 65.75 seconds.

It's not the one with shortest journey time is used for allocation. If the shortest waiting time is used, for example, 6.54s, it will take much longer time for the other calls to wait for, so total waiting time (cost) will be much larger. That is the linear programming's assignment problem.

4.6 Typical Assignment Problems in the lift call allocation

The problem discussed above is a special case, the typical problem in the lift system is that some times the number of calls is more than that of lifts; some times, it's less. For these cases, the assignment problem can be very flexible to deal with.

Dummy concept is introduced if the number of calls is not equal to that of lifts, i.e., $m \neq n$. According to the rules of assignment, the number of lifts must equal the number of calls, i.e., $m = n$, so the additional difference of $(m\text{-}n)$ calls or $(n\text{-}m)$ lifts must be introduced into system, this additional calls or lifts are called *dummy elements*. The role of these dummy calls or systems is to provide the fictional second calls to lifts, or fictional second lifts to calls. There are no costs for producing the fictional calls or lifts, so the cost entries for the dummy tasks are 0. The only exception is that the lift can't be used to respond any call or certain call or reserved by somebody.

4.6.1 *Number of calls less than the number of lifts*

Under normal operation, except for up peak and down peak, there are few calls in the system, so, the number of calls is normally less than that of lifts during a short interval. This problem seems quit straight forward, the responding speed is quit faster than the other case, because there have enough lifts to be free one. If the system uses the probability or forecast function, during certain period, the car will be pre-assigned to the call[(3)]. Here we only discuss the normal operation.

Take the other example as following, the calls now are only 3. 2 more dummy calls are added to meet the requirement of assignment.

Table 6: Response time of each lift to each call (s)							
Lift Status			**Call floor & Type**				
Lift	Position	Direction	21D	16U	4D	C(D1)	C(D2)
L1	10	Up	30.34	15.34	25.89	0	0
L2	2	Down	54.49	43.49	17.09	0	0
L3	15	Down	25.89	14.89	26.34	0	0
L4	7	Up	36.94	21.94	19.29	0	0
L5	19	Up	10.54	19.29	45.69	0	0

The smallest element in this table is 0, subtract 10.54, 14.89 and 17.09 from column 21D, 16U and 4D, we get following table.

Table 7: Response time of each lift to each call (s)							
Lift Status			**Call floor & Type**				
Lift	Position	Direction	21D	16U	4D	C(D1)	C(D2)
L1	10	Up	19.80	0.45	8.80	0	0
L2	2	Down	43.95	32.95	0✓	0	0
L3	15	Down	15.35	0✓	9.25	0	0
L4	7	Up	26.4	7.05	2.2	0	0
L5	19	Up	0✓	4.4	28.6	0	0

We can easily find 3 assignments to 0 element position possible. That's the solution. If the possible solution can't be found, go on the step 3, until it becomes possible.

So, the resulting total cost for this problem is

$Z = c_{23}+c_{32}+c_{51}$=17.09+14.89+10.54=42.52 seconds

The optimal solution for this assignment problem is: Lift 1 to dummy call; Lift 2 to storey 4 down call; Lift 3 to storey 16, up call; Lift 4 to dummy call; Lift 5 to storey 21, down call.

4.6.2 *Number of calls greater than the number of lifts*

This problem seems more difficult than the above one, because the number of calls compared to the number of lifts is quit big. For example, the case we used to simulate has 22 storeys and 5 lifts. So, actually the number of up and down calls is 42. Number of lifts is only 5. During the peak time, when calls are made, one lift may deal with several calls during one trip is normal. Because the lifts are almost equally assigned to response the calls, so the chances for 5 lifts are equal. If actually 42 calls registered in the system, according to the rule of assignment, there must have 42 lifts to perform the tasks. Because the chance for each lift is equal, each lift will have 9 calls average to be assigned, so there will be 45 lifts to 45 calls (3 dummy calls added here to keep the balance of calculation).

Here is the example, the cost of the system is the responding time of each lift to each call. Suppose the lifts have the equal weight to response calls, here, 2 dummy calls added.

Table 8: Response Time of Each Lift to Each Call (in seconds)

Lift Status			Call floor & Type							
Lift	Position	Direction	21D	16U	12D	9U	8D	4D	C(D1)	C(D2)
L1	10	Up	30.34	15.34	10.54	14.89	17.09	25.89	0	0
L1	10	Up	30.34	15.34	10.54	14.89	17.09	25.89	0	0
L2	2	Down	54.49	43.49	34.69	28.09	25.89	17.09	0	0
L2	2	Down	54.49	43.49	34.69	28.09	25.89	17.09	0	0
L3	15	Down	25.89	14.89	8.74	15.34	17.54	26.34	0	0
L3	15	Down	25.89	14.89	8.74	15.34	17.54	26.34	0	0
L4	19	Up	10.54	19.29	28.09	34.69	36.89	45.69	0	0
L4	19	Up	10.54	19.29	28.09	34.69	36.89	45.69	0	0

Use the assignment rule, subtract 10.54, 14.89, 8.75, 14.89, 17.09 and 17.09 from each column, we get the following table.

Table 9: Response Time of Each Lift to Each Call (in seconds)

Lift Status			Call floor & Type							
Lift	Position	Direction	21D	16U	12D	9U	8D	4D	C(D1)	C(D2)
L1	10	Up	19.8	0.45	1.8	0✓	0*	8.8	0	0
L1	10	Up	19.8	0.45	1.8	0*	0✓	8.8	0	0
L2	2	Down	43.95	28.6	25.95	13.2	8.8	0✓	0	0
L2	2	Down	43.95	28.6	25.95	13.2	8.8	0*	0	0
L3	15	Down	15.35	0✓	0*	0.45	0.45	9.25	0	0
L3	15	Down	15.35	0*	0✓	0.45	0.45	9.25	0	0
L4	19	Up	0✓	4.4	19.35	19.8	19.8	28.6	0	0
L4	19	Up	0*	4.4	19.35	19.8	19.8	28.6	0	0

From the table, the 6 assignments to 0 element position are possible.

Many ways to make a complete assignment. One of them is by 6 ✓. The resulting total cost is seem from table-8 to be:

$$Z = c_{14}+c_{25}+c_{36}+c_{52}+c_{63}+c_{71} = 10.54+14.89+8.74+14.89+17.09+17.09=83.24 \text{ seconds}$$

The total cost with * is the same as above.

The optimal solution for this assignment problem is: Lift 1 to storey 9, up call and storey 8, down call; Lift 2 to storey 4 down call; Lift 3 to storey 16, up call, storey 12 down call; Lift 4 to storey 21, down call.

4.7 Rules on lift allocation

The allocation is based on the moment on which the hall call is registered, that is, if there has one call made; the linear programming will allocate the lift to answer it. This allocation can be changed if there has other call registered before the lift reaching the destination, and the lift is still in full speed. If the lift is in slow down stage, so, at this time, this lift is deemed not to answer the new call. If the lift is running under full speed, when a car call is made after the passing storey, it will not be registered. If the call is between the destination and present position, and the distance is more than 2 times of that of slow down and speed up distance required. Then it can be answered.

If the waiting time of one call has reached designated period, one lift will be assigned to pick it up, during this period, this lift and call will be blocked from the allocation algorithm.

5. Conclusion

This system is based on the principle of quality service. The purpose of this is to minimize the total waiting time of each call and to enhance the system's performance. The result of this is based on the expense of the total of the journey times of all the cars.

6. ACKNOWLEDGEMENT

The author would like to thank Mr. Quah Eng Hing for his support on this paper and providing a lot reference books.

7. BIOGRAPHY

The author graduated at 1986, then worked as electrical research and design engineer for more than 10 years in Suzhou Schindler Elevator Company Ltd. In 1997, he became the member of IAEE, in the same year, He joined Chevalier Singapore Holdings Limited as technical service engineer. Email: Guifeng_chen@yahoo.com

8. REFERENCES :

(1) Artificial Neural Networks in Elevator Dispatching, B.A. Powell, Elevator Technology 10, Proceedings of ELEVCON 2000.
(2) G.C.Barney, S.M. dos Santos, Elevator Traffic Analysis Design and Control, 2ed, 1985.
(3) Improvement of Single Elevator Operation by Using Prior Standby Control, K. Yamashita, etc., Elevator Technology 10, Proceedings of ELEVCON 2000.
(4) System Analysis, E6203 Lecture Notes, EEE/NTU, Singapore, 1999/2000.