Decision Making in Manufacturing and Services
Vol. 1 • 2007 • No. 1–2 • pp. 175–190

Elevator Trip Distribution for Inconsistent Passenger Input-Output Data

Kiyoshi Yoneda*

Abstract. Accurate traffic data are the basis for group control of elevators and its performance evaluation by trace driven simulation. The present practice estimates a time series of inter-floor passenger traffic based on commonly available elevator sensor data. The method demands that the sensor data be transformed into sets of passenger input-output data which are consistent in the sense that the transportation preserves the number of passengers. Since observation involves various behavioral assumptions, which may actually be violated, as well as measurement errors, it has been necessary to apply data adjustment procedures to secure the consistency. This paper proposes an alternative algorithm which reconstructs elevator passenger origin-destination tables from inconsistent passenger input-output data sets, thus eliminating the *ad hoc* data adjustment.

Keywords: traffic problems, inverse problems, estimation, operation research
Mathematics Subject Classification: 96B20, 15A29
Received/Revised: 20 January 2007/28 June 2007

1. INTRODUCTION

This paper concerns acquisition of traffic data for elevator systems. The data acquired are used for *elevator group control* (Nikovski, Brand, 2003) and its evaluation by *trace driven simulation* (Smith, 1994; Uhlig, 1997), both explained briefly in Appendixes A and B. The desired data are, for each elevator passenger, (a) arrival time, (b) origin floor, and (c) destination floor. Hopefully these data are to be monitored continuously on a real time basis without requiring special data acquisition equipment. Since none of the data items is automatically available, the desired data need to be reconstructed from sensor data such as elevator load, buttons pressed, and elevator location.

When an elevator car arrives at a floor, first some of the passengers go out of the car; only upon its completion passengers in the elevator hall begin to flow into the car. Hence the number of passengers who went out of and who came into the car may be measured fairly accurately by observing the change of load from the arrival to the departure of the car:

$$\begin{aligned} \text{load\_out} &= \text{load\_at\_arrival} - \text{minimum\_load\_at\_the\_floor} \\ \text{load\_in} &= \text{load\_at\_departure} - \text{minimum\_load\_at\_the\_floor} \end{aligned}$$

* Fukuoka University, Japan. E-mail: yoneda@econ.fukuoka-u.ac.jp.

Some of the passengers get off the car at a floor only temporarily to make way to passengers wishing to exit the car. This number is large when the car is crowded and small when not. Some passengers are heavy while others are light. By compensating for these by various means it is possible to obtain an estimate of the *input-output (IO) data* which consists of the number of passengers who departed from and the number of passengers who arrived at the particular floor by the elevator car under observation.

A unit of elevator car observation starts at the moment the car becomes empty, proceeds through a process of passengers going out of and coming into the car as the car makes stops, ending when the car becomes empty again. Such a unit of observations is called a *regenerative* data set after a terminology in simulation (Ross, 1996). We assume that an elevator car becomes empty at least when the car changes the direction from up to down or vice versa. (Actually this does not hold when the system is so crowded that passengers ride any elevator car that stops, disregarding its direction.) By distributing the IO data to an *origin-destination (OD) table* we have reconstructed data items (a), (b), and (c), as will be detailed later in this paper: now we know how many passengers were carried from which floor to which floor and when that event took place.

Since the observation described involves not only various behavioral assumptions regarding the passengers but also many sources of measurement errors, the regenerative data set is prone to be *inconsistent*, meaning that the estimated number of passengers may differ between the total departure and the total arrival. This complicates the downstream data processing, viz. the distribution of regenerative IO data into an OD table. For this reason it has been customary to perform a preprocessing to ensure that the total number of departing passengers equals the total number of arriving passengers within a regenerative data set. The schematic data flow would then be

$$\begin{aligned}\text{Regenerative data set} &\rightarrow \text{(Inconsistent) IO data} \\ &\rightarrow \text{Consistent IO data} \rightarrow \text{OD table.}\end{aligned} \tag{1}$$

This paper documents the present status of elevator trip observation technology and then proposes a new method to distribute regenerative IO data into an OD table without requiring the *ad hoc* adjustment to enforce consistency:

$$\text{Regenerative data set} \rightarrow \text{(Inconsistent) IO data} \rightarrow \text{OD table.} \tag{2}$$

Each regenerative data set obtained, to be described in Section 2, is then transformed into IO data as in Section 3. The present industrial practice to enforce data consistency by preprocessing and then distribute the IO data into an OD table is explained in Section 4. The original contribution of this paper is in Section 5 where the inconsistent IO data are directly distributed into an OD table. Section 6 illustrates that skipping the IO data reconstruction as in

$$\text{Regenerative data set} \rightarrow \text{OD table} \tag{3}$$

is conceivable but impractical. The results are summarized in Section 7 with comments. Appendixes A and B include background information on elevator group control and trace driven simulation, respectively.

2. REGENERATIVE DATA SETS

A cheap way to monitor traffic of a large number of elevators is to use sensors that elevators readily have and send collected data to a computer over a communication channel. This is not difficult since modern elevators come with remote monitoring systems for safety and maintenance purposes. Available sensors vary among elevator systems but usually include those listed in Table 1.

Table 1. Commonly available elevator car status information

Current time	
Direction	neutral, heading up, heading down
Location	between floors i and j, $j = i$ or $j = i + 1$
Load	weight onboard
Door status	open or closed
Hall button status	up/down call pressed or not
Car button status	destination k pressed or not

Additional data are sometimes available such as door control button signals, photocell door sensor signals, or even images from monitoring cameras. However, development of data collection methods specific to those exceptional systems is seldom justifiable.

Suppose the observation of a car starts at the moment it becomes empty. After some time passengers will walk into the elevator hall, press the hall button, and get in when the car arrives. The car travels upward or downward; some passengers get off and others get on as the car makes stops. Eventually the car becomes empty again. We call such a sequence of observations beginning and ending with null load a *regenerative* set of data following the terminology in simulation (Ross,1996).

Example 1. Regenerative data set.

In this example from (Yoneda et al. 1997), Figures 1 to 4 illustrate a sequence of events from which a regenerative data set is acquired.

The elevator car is initially empty at stop 1 (not necessarily floor 1) as in Figure 1. When the car leaves stop 1 its load corresponds to five passengers. Car buttons to stops 3, 4, and 5 (not necessarily floors 3, 4, and 5) have been pressed prior to the arrival at stop 2, and that at stop 2 the upward hall button has been pressed. The car proceeds upward to stop 2, at which no passenger gets off.

At stop 2, one passenger gets into the elevator car, as illustrated in Figure 2. When the car leaves stop 2 its load is six passengers with the same car buttons pressed as before. Upward hall button has been pressed at stop 3. Upon arrival at stop 3, the car button to stop 3 is cleared; two passengers get off.

At stop 3, two passengers ride the elevator car as in Figure 3. When the car leaves stop 3, its load is six again. Upon arrival at stop 4, the car button to stop 4 is cleared; four passengers get off.

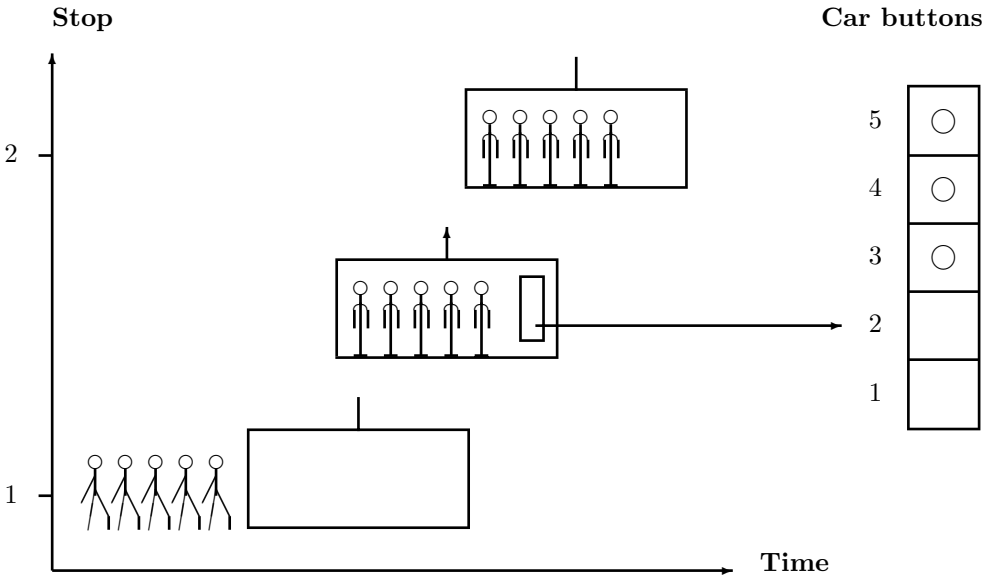

Fig. 1. From stop 1 to stop 2

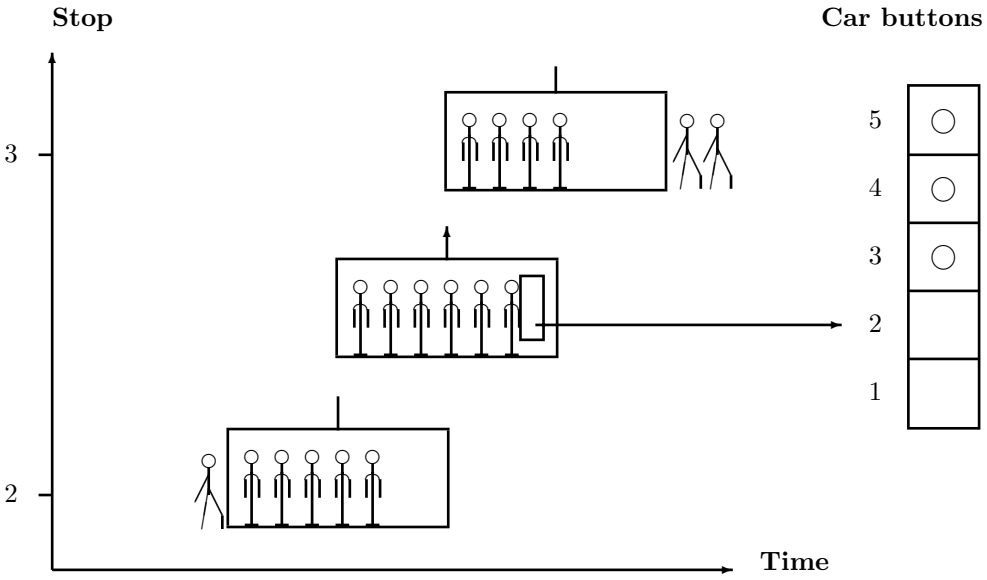

Fig. 2. From stop 2 to stop 3

At stop 4, no new passenger rides the elevator car. When it leaves stop 4, the load is two as in Figure 3. At floor 5, the car becomes empty again; a regenerative data set has been acquired.

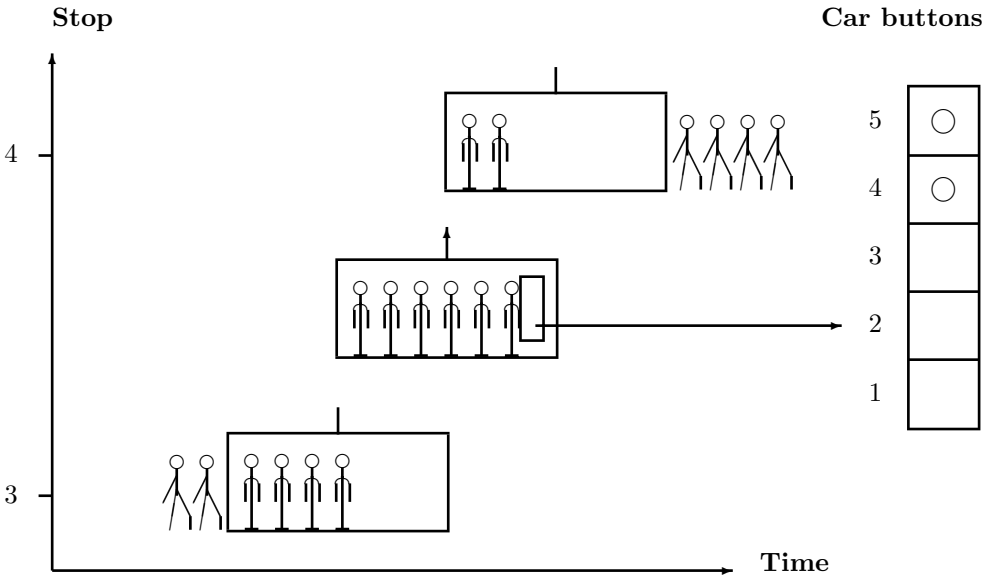

Fig. 3. From stop 3 to stop 4

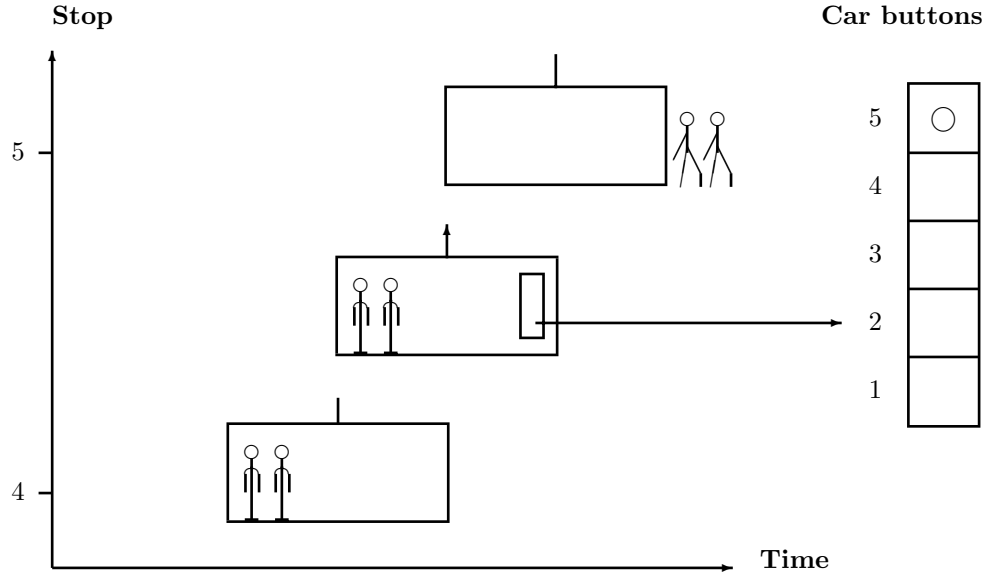

Fig. 4. From stop 4 to stop 5

The entire process is summarized in Figure 4. Note that the exact number of passengers in the car is unknown since it is only estimated from the weight measurement of the load.

With such a regenerative set of data it is desired to estimate how many passengers traveled from stop i to stop j, for all stop pairs (i, j),

3. PASSENGER IO DATA

As has been outlined in Section 1, an estimate of the number of passengers coming out of and going into an elevator car at a stop may be obtained from elevator car loads.

The arrival time of a passenger can be narrowed down to the interval between the time the hall button of the relevant direction was pressed and the time the elevator car of the same direction arrived. At least one passenger is sure to have arrived at the moment the hall button was pressed (unless the passenger leaves the elevator hall without riding the elevator, which is not very likely). The rest of the arrivals can be estimated according to the arrival process model, such as Poisson arrival or some kind of group arrival.

To summarize, with a regenerative set of data,

1. The elevator load carried from one stop i to the next $i+1$ can be measured fairly accurately.
2. At a stop i it is possible to estimate $n_{\cdot i}$, the number of passengers which the elevator carried *to* stop i, and $n_{i\cdot}$, the number of passengers which the elevator carried *from* stop i.
3. The arrival time of the first passenger can be measured fairly accurately for each (floor, direction)-pair.
4. The arrival times of the other passengers may be estimated according to the arrival process model.

However, since the estimates are based upon passengers' *behavioral assumptions* which may well be violated and also subject to measurement errors (such as the "hysteresis" of the scale), the data set is usually inconsistent meaning that

$$\sum_i n_{i\cdot} = \sum_j n_{\cdot j} \tag{4}$$

may not hold, as if the number of passengers were not conserved by transportation using elevators.

4. IO DATA WITH ENFORCED CONSISTENCY

For trace driven simulation of elevator groups traffic has to be supplied for all floor pairs. Since there is no direct way to measure the traffic between floor pairs, it has to be reconstructed mainly from item 2 in the previous section. The industrial practice has been to make a preliminary adjustment to satisfy the consistency (4) in order to facilitate the downstream data processing.

The easiest way to go is (1), to normalize $n_{i\cdot}$ and $n_{\cdot j}$, for instance, by $n_{\cdot\cdot}\, n_{i\cdot}/\sum_k n_{k\cdot}$ and $n_{\cdot\cdot}\, n_{\cdot j}/\sum_\ell n_{\cdot\ell}$, respectively, where $n_{\cdot\cdot}$ is something like, say, the average of $\sum_k n_{k\cdot}$ and $\sum_\ell n_{\cdot\ell}$ rounded to an integer.

Example 2. Consistent data from Figure 5.

Assuming that the numbers of passengers who went into and came out of the elevator car at each stop are accurately known, the formulation into *origin-destination (OD) table* estimation proceeds as follows.

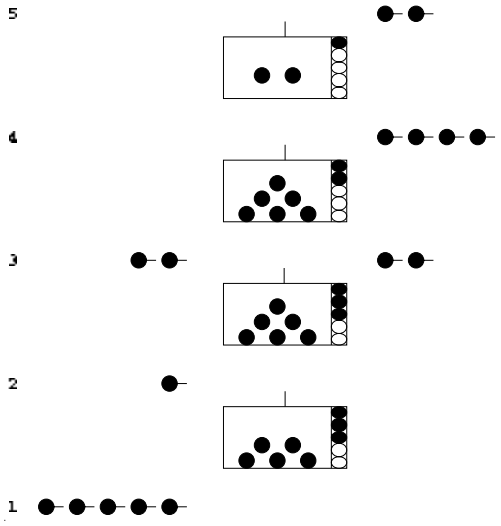

Fig. 5. A regenerative data set

The question marks "?" are for the unknowns while the blanks are for zero:

OD	1	2	3	4	5	$\sum$
1			?	?	?	5
2			?	?	?	1
3				?	?	2
4						
5						
$\sum$			2	4	2	8

Since buttons to stops 3, 4, and 5 have been pressed prior to the car's arrival at stop 2, it may be concluded that at least one person moved from stop 1 to each of the stops 3, 4, and 5:

OD	3	4	5	$\sum$
1	?	?	?	2
2	?	?	?	1
3	0	?	?	2
$\sum$	1	3	1	5

This table enables the use of various *matrix balancing algorithms*, notably the easy-and-fast RAS algorithm (Schneider, Zenios, 1990). The RAS algorithm with the uniform prior adding up to 5

$$\begin{bmatrix} 5/8 & 5/8 & 5/8 \\ 5/8 & 5/8 & 5/8 \\ 0 & 5/8 & 5/8 \end{bmatrix}$$

results in:

OD	3	4	5	$\sum$
1	0.667	1.000	0.333	2
2	0.333	0.500	0.167	1
3	0	1.500	0.500	2
$\sum$	1	3	1	5

The only remaining problem is that the solution thus obtained is in real numbers rather than integers. Rounding the reals into integers results in inconsistencies as follows

OD	3	4	5	$\sum$
1	1	1	0	2
2	0	1	0	1
3	0	2	1	*2*
$\sum$	1	*3*	1	5

□

Consistent integer estimates may be obtained by the *greedy algorithm* which assumes that the passengers arrived in the order of likelihood. The algorithm is much faster than trying to obtain an integer solution directly as in Yoneda (1994).

Example (continued). Integer solution.

Since the most probable arrival is a passenger going from stop 3 to stop 4, assume that such a passenger actually existed.

Then we have

OD	3	4	5	$\sum$
1	0	0	0	0
2	0	0	0	0
3	0	1	0	1
$\sum$	0	1	0	1

and the OD table for the remaining passengers is

OD	3	4	5	$\sum$
1	?	?	?	2
2	?	?	?	1
3	0	?	?	1
$\sum$	1	2	1	4

Solving this yields

OD	3	4	5	$\sum$
1	0.667	0.889	0.444	2
2	0.333	0.444	0.222	1
3	0	0.667	0.333	1
$\sum$	1	2	1	5

from which we have the next arrival $(1, 4)$

OD	3	4	5	$\sum$
1	0	1	0	1
2	0	0	0	0
3	0	1	0	1
$\sum$	0	2	0	2

with the OD table for the remaining passengers

OD	3	4	5	$\sum$
1	?	?	?	1
2	?	?	?	1
3	0	?	?	1
$\sum$	1	1	1	3

Proceeding similarly the greedy algorithm terminates with

OD	3	4	5	$\sum$
1	1	1	0	2
2	0	1	0	1
3	0	1	1	2
$\sum$	1	3	1	5

We have secured a reconstructed OD table. □

5. INCONSISTENT IO DATA

We now drop the assumption that the IO data are consistent to adopt (2). Since the RAS algorithm oscillates when the data are inconsistent, an alternative algorithm

is required. Various candidates exist, including the *network optimization algorithm,* also described in Schneider, Zenios (1990), which we find to be too involved. Here we choose the *positive inverse problem* approach in Yoneda (2006) for simplicity. The method finds a unique positive solution to approximate a generally inconsistent system of positive near-equations of the form

$$\begin{bmatrix} 1 & & \\ & \ddots & \\ & & 1 \\ & \vdots & \\ \cdots & X_{i\,j} & \cdots \\ & \vdots & \end{bmatrix} \begin{bmatrix} \vdots \\ \theta_j \\ \vdots \end{bmatrix} \approx \begin{bmatrix} \vdots \\ \vdots \\ \vdots \\ y_i \\ \vdots \end{bmatrix} \qquad w = \begin{bmatrix} \vdots \\ \vdots \\ \vdots \\ w_i \\ \vdots \end{bmatrix}$$

$$0 \le X_{ij},\ 0 < \theta,\ 0 < y,\ 0 \le w$$

where w is a vector of importance weights. Note that this is an interesting problem in its own right since most numbers such as prices and quantities which appear in various real-world applications are positive. Zero has been excluded as a possible value of θ to avoid complications related to the difference between estimation and model identification.

The solution method proposed in Yoneda (2006) is as follows. Write the system of near-equations $X\theta \approx y$, or equivalently $X_{i\cdot}\theta \approx y_i$, where $X_{i\cdot}$ is the i-th row vector of X. By writing $Z_{ij} := X_{ij}/y_i$ the system may be rewritten as $Z\theta \approx 1$. Noting that

$$Z_{i\cdot}\theta - 1 = 0 \quad \Leftrightarrow \quad \log Z_{i\cdot}\theta = 0$$
$$0 \le (Z_{i\cdot}\theta - 1)\log Z_{i\cdot}\theta$$

the solution to $Z\theta \approx 1$ is defined by

$$\hat{\theta} := \arg\min_{\theta} \sum_i (Z_{i\cdot}\theta - 1)\log Z_{i\cdot}\theta \ .$$

Since this is minimization of a strictly convex smooth function the solution is fast and easy. The computational effort required is greater than the RAS algorithm (the greater the inconsistency, the more the effort needed) but well within the capacity of personal computers or even elevators' onboard microcomputers.

Example 3. Inconsistent IO data.

Suppose that in the previous example the measured number of passengers who went into the elevator does not match the number of passengers who came out, as follows:

OD	3	4	5	$\sum$
1	θ_1	θ_2	θ_3	2
2	θ_4	θ_5	θ_6	1
3	0	θ_7	θ_8	2
$\sum$	1	2	1	4 or 5?

Then we have, say,

$$\begin{bmatrix} 1 & & & & & & & \\ & 1 & & & & & & \\ & & 1 & & & & & \\ & & & 1 & & & & \\ & & & & 1 & & & \\ & & & & & 1 & & \\ & & & & & & 1 & \\ & & & & & & & 1 \\ 1 & 1 & 1 & & & & & \\ & & & 1 & 1 & 1 & & \\ & & & & & & 1 & 1 \\ 1 & & & 1 & & & & \\ & 1 & & & 1 & & 1 & \\ & & 1 & & & 1 & & 1 \end{bmatrix} \begin{bmatrix} \theta_1 \\ \theta_2 \\ \theta_3 \\ \theta_4 \\ \theta_5 \\ \theta_6 \\ \theta_7 \\ \theta_8 \end{bmatrix} \approx \begin{bmatrix} 4.5/8 \\ 4.5/8 \\ 4.5/8 \\ 4.5/8 \\ 4.5/8 \\ 4.5/8 \\ 4.5/8 \\ 4.5/8 \\ 2 \\ 1 \\ 2 \\ 1 \\ 2 \\ 1 \end{bmatrix} \qquad w = \begin{bmatrix} 1/100 \\ 1/100 \\ 1/100 \\ 1/100 \\ 1/100 \\ 1/100 \\ 1/100 \\ 1/100 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \end{bmatrix}$$

where the weights indicate that the priors are one digit less credible than the IO data, i.e., 1/10 in standard deviation or 1/100 in variance. Its solution gives

OD	3	4	5	$\sum$
1	0.703	0.702	0.350	1.755
2	0.364	0.363	0.220	0.947
3	0	1.217	0.512	1.729
$\sum$	1.067	2.282	1.086	4.431

so that the first arrival is

OD	3	4	5	$\sum$
1	0	0	0	0
2	0	0	0	0
3	0	1	0	1
$\sum$	0	1	0	1

Proceeding similarly to the greedy algorithm with consistent data we arrive at

OD	3	4	5	$\sum$
1	1	0	0	1 (< 2)
2	0	1	0	1
3	0	1	1	2
$\sum$	1	2	1	4

□

With inconsistent data the greedy algorithm always favors the minimum of the row and column sums. It is easy to modify the algorithm so that the total estimated number of passengers equals, say, the closest integer to the average of row and column sums. It is also easy to set a higher weight to the more credible between the row and the column sums.

6. SKIPPING IO DATA

Up to the previous section we first documented the industrial practice (1) and then proposed a way to proceed more directly as in (2). The advantage of permitting inconsistent IO has been that the method requires no preprocessing to ensure data consistency, unlike with the traditional method. Going a step further, now we drop the computation of IO data, which is likely to be inaccurate anyway, to adopt (3). This is possible since the method proposed in the previous section solves any system of positive linear approximate equations. However, if we assume that only load carried from a stop to another is available in place of the IO data, information seems insufficient even for the simple example used in this paper:

Example 4. Load data without IO.

If IO data are unavailable, the near-equations become, say,

$$\begin{bmatrix} 1 & & & & & & & \\ & 1 & & & & & & \\ & & 1 & & & & & \\ & & & 1 & & & & \\ & & & & 1 & & & \\ & & & & & 1 & & \\ & & & & & & 1 & \\ 1 & 1 & 1 & & & & & \\ 1 & 1 & 1 & 1 & 1 & 1 & & \\ & 1 & 1 & & 1 & 1 & 1 & 1 \\ & & 1 & & & 1 & & 1 \end{bmatrix} \begin{bmatrix} \theta_1 \\ \theta_2 \\ \theta_3 \\ \theta_4 \\ \theta_5 \\ \theta_6 \\ \theta_7 \\ \theta_8 \end{bmatrix} \approx \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 2 \\ 3 \\ 3 \\ 2 \end{bmatrix} \qquad w = \begin{bmatrix} 1/100 \\ 1/100 \\ 1/100 \\ 1/100 \\ 1/100 \\ 1/100 \\ 1/100 \\ 1/100 \\ 1 \\ 1 \\ 1 \\ 1 \end{bmatrix}$$

where the weights for priors are taken even smaller than before since not even the total number of passengers is now known. The solution is

OD	3	4	5	$\sum$
1	0.678	0.447	0.833	1.958
2	0.383	0.281	0.447	1.111
3	0	0.384	0.675	1.062
$\sum$	1.062	1.111	1.958	4.131

so that the first arrival is $\theta_3 = 1$:

OD	3	4	5	$\sum$
1	0	0	1	1
2	0	0	0	0
3	0	0	0	0
$\sum$	0	0	1	1

The next near-equations are then

$$\begin{bmatrix} 1 & & & & & & & \\ & 1 & & & & & & \\ & & 1 & & & & & \\ & & & 1 & & & & \\ & & & & 1 & & & \\ & & & & & 1 & & \\ & & & & & & 1 & \\ 1 & 1 & 1 & & & & & \\ 1 & 1 & 1 & 1 & 1 & 1 & & \\ & 1 & 1 & & 1 & 1 & 1 & 1 \\ & & 1 & & & 1 & & 1 \end{bmatrix} \begin{bmatrix} \theta_1 \\ \theta_2 \\ \theta_3 \\ \theta_4 \\ \theta_5 \\ \theta_6 \\ \theta_7 \\ \theta_8 \end{bmatrix} \approx \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 2 \\ 2 \\ 1 \end{bmatrix} \qquad w = \begin{bmatrix} 1/100 \\ 1/100 \\ 1/100 \\ 1/100 \\ 1/100 \\ 1/100 \\ 1/100 \\ 1/100 \\ 1 \\ 1 \\ 1 \\ 1 \end{bmatrix}$$

Although it is possible to proceed by greedy algorithm as in previous examples, it becomes apparent that even with this example the near-equations do not have enough information so that the assignment of a person to the OD pair becomes arbitrary, viz., the integer solution will not be unique. For instance, these two solutions are equally likely:

OD	3	4	5	$\sum$
1	2	0	1	3
2	0	0	0	0
3	0	1	0	1
$\sum$	2	1	1	4

OD	3	4	5	$\sum$
1	1	0	1	2
2	0	1	0	1
3	0	1	0	1
$\sum$	1	2	1	4

□

7. CONCLUDING REMARKS

A new elevator trip distribution method has been proposed, which has the following properties:

— The *ad hoc* data adjustment preprocessing, which has been a common practice, is eliminated.

— The computational effort is slightly greater than the RAS algorithm but well within the commonly available computational capabilities.

The first experiment by Toshiba Corporation to obtain regenerative data sets was conducted 1975 in Tokyo after the author's suggestion[1]. Inspection of patents applied

[1] The first measurement equipment was the size and weight of a refrigerator because it comprised microcomputer boards and recording devices. It was a hard work to pull it up to the elevator machine room at the top floor, since the elevator does not go to the floor hosting its motor-generator.

for indicate that other Japanese elevator manufacturers may have conducted similar experiments around that time as well.

It should be possible to improve the accuracy of traffic estimation by introducing more sensor data such as a monitoring camera in each elevator car. Although that would certainly increase the credibility of trace driven simulations, it is doubtful if the same would result in an indisputably better elevator group control. This is not only because the concept of elevator efficiency is obscure as pointed out in Appendix A but also because the present elevator configuration which restricts the operation to a single car per elevator shaft severely limits the traffic throughput.

The idea of operating multiple cars in two elevator shafts, one upward and the other downward, has been in the air for at least half a century. With the advent of linear motors and inexpensive microcomputers no serious engineering obstacle seems to exists in implementing the idea. Such systems bring about many new problems regarding its operation. For instance, given expected inter-floor traffic, how many elevator cars should be deployed in a pair of upward and downward elevator shafts? What size should an elevator car be? Academic research directed towards such future elevator systems would seem more productive than those aiming to squeeze out small increase in utility from traditional elevator systems while it is perfectly understandable that industrial research need results immediately applicable to existing systems.

Acknowledgments

The author is grateful to Mr. Hiroshi TAKEUCHI, who was with Toshiba Fuchu Works when the author worked on elevator projects in the late 1970s, for being permissible to what was then an unconventional idea.

Appendixes

A. ELEVATOR GROUP CONTROL

In high-rise buildings elevators are divided into *groups* belonging to the same elevator bank and facing the same elevator hall on all serving floors. Aiming to achieve a higher efficiency by coordinating them, *elevator group control* has been researched by elevator manufacturers. See Nikovski, Brand (2003) for a summary of the technology and literatures available in English.

The difficulty of elevator group control lies not so much in optimizing a given objective function but rather in agreeing on a reasonable objective function.

In academia attention is often focused on minimizing the passengers' average waiting time. This is, however, not unlike saying that all that matters in a computer system is the average response time. If the average waiting time were the only issue, an elevator car moving upward would likely do better by fetching not only passengers

going up but also those wishing to go down since the time the passengers waste in the car does not count. Adopting passengers' average system time instead of average waiting time does not change the situation significantly because it is usually more efficient to stop only once going up instead of twice, once when going up and again when coming down. The fact that no group controlled elevator exhibits such behavior suggests that factors other than those simple performance indexes are important. Certain behavior patterns are expected of elevators; those violating them are deemed unacceptable.

If elevator group control aims at reducing complaints from passengers the average waiting time is inappropriate as a performance index since a passenger who had to wait a long time is more likely to complain than a passenger who had to wait a near-average waiting time. For this reason Mitsubishi states the minimization of a quadratic function of waiting time to be an important element in their objective function favoring short-tailed waiting time distribution in detriment of the average.

The choice between *reassignment* and *immediate* policies also illustrates that waiting time is not the only issue. Japanese elevator companies adopt immediate policy meaning that when a passenger presses a hall button the assigned car is announced to the passenger at once and remains unchanged until it arrives. The passengers who wish to take that car walk up to its door and wait paying no further attention to the behavior of the other elevators. On the other hand, reassignment is the rule in the West meaning that car assignment may change depending on circumstances. This shortens passengers' average waiting time but requires that they beware of changes in schedule. Deciding which one provides a better service to the passengers is not an easy task even when a numerical description of the tradeoff is available, which suggests another criterion to settle the issue: manufacturers may adopt immediate policy not so much to provide a better service but more to simplify the control logic.

In a wider perspective, passengers are not the only customers to take care of. For instance, under low traffic the building owner may prefer to turn off the motor-generator for some of the elevator cars to save energy even if the level of customer service drops as a consequence. In order to take a multitude of elements in consideration manufacturers such as Otis and Toshiba adopt weighted sums of various performance indexes as the objective function, which may be considered a kind of multiple attribute utility function.

B. TRACE DRIVEN SIMULATION

All these suggest that a good elevator group control means well-balanced operation rather than optimization of a simple performance criterion. A natural adaptation strategy for an elevator manufacturer in this situation is to leave key parameters, such as weights in the sums, adjustable and present simulation results to the customers to show tradeoffs among performance indexes. The customers may decide on the parameters for themselves after having seen the simulation results. To make the simulation persuasive it is necessary that the input data be realistic, not something idealized like Poisson arrival to the entrance hall with a uniformly distributed destination floors.

Traffic data should be based on real observation obtained from a building similar to the one under study. Such form of simulation, called *trace driven simulation* (Smith, 1994; Uhlig, Mudge, 1997), has been popular in performance analyses of computer and communication systems. A major issue in trace driven simulation is how to obtain a large amount of data at a low cost, which is the problem dealt with in the present paper.

REFERENCES

1. D. Nikovski, M. Brand, *Decision-theoretic group elevator scheduling*, in *International Conference on Automated Planning and Scheduling (ICAPS)*, Mitsubishi Electric Research Laboratories, 2003, number TR2003-061 in Technical Report.
2. S. M. Ross, *Simulation*, Academic Press, second edition, 1996.
3. M. H. Schneider, S. A. Zenios, *A comparative study of algorithms for matrix balancing*, Operations Research, 38 (3), (1990), 439–455, ISSN 0030-364X.
4. A. J. Smith, *Trace driven simulation in research on computer architecture and operating systems*, in S. Morito *et al.*, editors, *New Directions in Simulation for Manufacturing and Communications*, The Operations Research Society of Japan, 1994, 43–49.
5. R. A. Uhlig, T. N. Mudge, *Trace-driven memory simulation: A survey*, ACM Computing Surveys, 29 (2), (1997), 128–170.
6. K. Yoneda, *Integer estimation of origin-destination tables*, Transactions of Institute of Electric Engineers of Japan, 114-C (4), (1994), 483–490.
7. K. Yoneda, *A parallel to the least squares for positive inverse problems*, Journal of the Operations Research Society of Japan, 49 (4), (2006), 279–289.
8. K. Yoneda, Y. Nakayama, T. Matsumoto, *Trace-driven simulation of elevator groups*, Communications of the Operations Research Society of Japan, 42 (5), (1997), 371–374. In Japanese.