The Effect of Randomization on Constraint Based Estimation of Elevator Trip Origin-Destination Matrices

Juha-Matti Kuusinen[1], Arnaud Malapert[2]

[1] KONE Corporation, 02150 Espoo, Finland, juha-matti.kuusinen@kone.com

[2] Université Nice Sophia Antipolis, 06903 Sophia Antipolis, France, arnaud.malapert@unice.fr

Keywords: elevator traffic, origin-destination matrix, constraint programming.

Abstract. We present a constraint programming formulation for the elevator trip origin-destination matrix estimation problem, and study different deterministic and randomized algorithms to solve the problem. An elevator trip consists of successive stops in one direction of travel with passengers inside the elevator. It can be defined as a directed network, where the nodes correspond to the stops on the trip, and the arcs to the possible origins and destinations of the passengers. The goal is to estimate the count of passengers for the origin-destination pairs of every elevator trip occurring in a building. These counts can be used to make passenger traffic forecasts which, in turn, can be used in elevator dispatching to reduce uncertainties related to future passengers. The results show that randomized search improves the quality of estimation results. In addition, the proposed approach satisfies real time elevator group control requirements for estimating elevator trip origin-destination matrices.

INTRODUCTION

Modern group controls typically plan elevator routes based on existing calls [1,2]. At any given moment, however, a passenger may arrive to an elevator lobby and give a new call which requires the changing of previously defined routes, if they are no longer optimal. By making forecasts of future passengers, the group control can avoid such unexpected route changes and improve passenger service level [3]. The forecasts should be based on complete information about the passenger traffic, i.e., on passenger journeys. A passenger journey is the journey of one passenger from an origin floor to a destination floor. The problem is that, especially during heavy traffic, the passenger journeys cannot be uniquely determined. They can, however, be estimated by solving the elevator trip origin-destination matrix (ETODM) estimation problem [4].

An elevator trip to up or down direction starts when passengers board an empty elevator and ends to a stop where the elevator becomes empty again. The passengers who board the elevator register calls that define their destinations, and the OD pairs of the trip. The boarding and alighting passenger counts can be obtained, e.g., by measuring stepwise changes in the load of an electronic load weighing device [5]. An estimated ETODM contains the OD passenger counts, i.e., the passenger journeys, for the OD pairs of the trip. The ETODMs estimated for a given time interval are added up to construct a building OD matrix (BODM) that describes the passenger traffic between every pair of floors in the building during that interval. The length of the time interval depends largely on the traffic intensity, but a typical interval is at least five and at most 15 minutes [6]. To learn the passenger traffic in the building, the BODMs of the same time of day or time interval, and usually day of week, are combined using, e.g., exponential smoothing [5]. The learned BODMs can be used to make forecasts about future passengers, namely, when and at which floors new passengers will register new calls, what is the number of passengers waiting behind the new and existing calls, and what are their destinations.

An elevator trip is analogous to a single transit route, e.g., a bus line, where there is only one route connecting any OD pair, and usually counts of the boarding and alighting passengers are collected on all stops on the route [7]. There are many methods for estimating the OD matrix for a single transit route. If the observed passenger counts are consistent, then a typical objective is to minimize a distance measure between the predicted and a target OD matrix subject to the so called flow conservation constraints. They simply require that passengers travelling on the route do not disappear or multiply. The target OD matrix is usually based on historical data or a survey. A popular distance measure is the information minimizing function [8]. Similar estimators are obtained with the iterative proportional fitting method [7,9,10,11], and recursive methods [12,13,14]. Other types of estimators are obtained with constrained generalized least squares (CGLS) and constrained maximum likelihood approaches [11]. If the observed boarding and alighting passenger counts are not consistent, then a distance measure between the predicted and observed counts should also be minimized. Popular approaches are the maximum likelihood, the Bayesian and the CGLS method [15,16,17,18,19].

A single transit route is usually defined in advance and remains as such for long periods of time. This means that it is possible to collect many counts on the same route during a given time period, e.g., a rush hour, and use these counts to estimate average passenger counts for the OD pairs of the route. An elevator trip is request driven which means that there may not be two similar elevator trips even within a day. In addition, every elevator trip has its own set of OD pairs, and boarding and alighting counts. This is why we need to estimate a separate OD matrix for each elevator trip. Because there cannot be partial passengers, only integer solutions are acceptable. If the requests and the measured counts affect the domain of the predicted OD passenger counts then, unlike in a single transit route, they must be taken into account when defining whether an ETODM estimation problem is consistent or not.

In [4], the ETODM estimation problem was formulated as a box-constrained integer least squares (BILS) problem and algorithms for finding all solutions to the problem were presented. When all solutions are available and one is selected every time, e.g., randomly or as the average of the solutions, the BODMs are not affected by the algorithm used in solving the problem. In the long term, this strategy results in BODMs that model better the possible realizations of the passenger traffic, and enable robust passenger traffic forecasting in elevator dispatching. In [20], an ETODM was estimated by solving a succession of positive inverse problems. Both of the above methods can solve inconsistent problems, but the latter finds only a single solution to the problem. This is a not a good property when the goal is to construct BODMs for passenger traffic forecasting. In [21], the ETODM estimation problem was formulated as a linear programming (LP) problem. The presented approach, however, can be used only for consistent problems.

For implementing an ETODM estimation algorithm in a real elevator group control application, the algorithm must be fast to reduce CPU load, and to have the most recent information about the passenger traffic all the time. The BILS approach is faster than the LP approach [4,21]. However, since the ETODM estimation problem is in general NP-hard, all solutions to sufficiently complex problems cannot be found within a reasonable time which in a real application can be defined to be at most 0.5 seconds.

We formulate the ETODM estimation problem as a constraint optimization problem (COP) [22]. The formulation is based on elevator movements, e.g., stops, service requests, e.g., landing and car calls, and counts of boarding and alighting passengers. In addition to respecting a set of constraints, a solution to the problem is optimal with respect to a predefined distance measure between the predicted and observed passenger counts. We selected the least squares (LS) objective function because it favors solutions where the difference between all of the predicted and observed counts is small, which is reasonable considering a real application.

One advantage of the CP approach compared to the previous approaches is that both deterministic and randomized optimization procedures, resulting in a single or multiple optimal solutions, can be easily implemented. Intuitively, if only some (instead of all) of the optimal solutions can be computed within a real time limit, then a randomized search should result in BODMs that describe better the possible realizations of the passenger traffic, i.e., BODMs of better quality. The reason is that a deterministic search will always favor particular solutions. By using different deterministic and randomized candidate algorithms (CA), we study the effect of randomization on BODM quality. BODM quality is measured based on the total squared deviation between the estimated and the true BODM. In addition, we compare the different CAs with respect to solving time.

CONSTRAINT PROGRAMMING FORMULATION

We define an elevator trip as a directed network of nodes $N = \{1,2,\dots,n\}$, and arcs A defined by OD pairs (i,j), $i,j \in N$. The node i corresponds to the ith stop on the elevator trip. Let r_i be the node at which a delivery request to the node $i \in N$, $r_i < i$, is registered. If no delivery requests are registered to node i, then $r_i = n+1$. Let b_i and a_i denote the measured count of passengers who board and alight at node $i \in N$, respectively. The elevator capacity, expressed as number of passengers, is denoted with C.

We assume that:

1. At any time, there are less than C passengers in the elevator.
2. At least one passenger boards at node $r_i \neq n+1$ and alights at node i.
3. Passengers do not alight at a node without a delivery request.
4. A passenger who boards at node $i < r_j$, i.e., before the delivery request to node j is registered, does not alight at node j.

The assumptions 2 and 3 imply that we trust the delivery requests. The fourth assumption means that the possible destinations of a passenger are defined by the delivery requests that are registered before or at the node where the passenger boards the elevator, which is usually the case in practice. This eliminates some OD pairs, and thus, an elevator trip often includes a smaller number of OD pairs than a single transit route where typically any node i forms an OD pair with any other node j, $i < j$.

The set of arcs A is defined as:

$$A = \left\{(i,j) \in N^2 \middle| i < j \wedge i \geq r_j\right\}. \tag{1}$$

Let $B_i \in [0,C]$ and $A_i \in [0,C]$ denote the predicted count of passengers who board and alight the elevator at node $i \in N$, respectively. Let $P_i \in [1,C]$, $i = 1,\dots,n-1$, denote the number of passengers in the elevator between the nodes i and $i+1$. Finally, let $X_{ij} \in [0,C]$ denote the predicted passenger count along the arc or OD pair $(i,j) \in A$, i.e., the passenger count from origin i to destination j, that we want to estimate.

The predicted boarding and alighting counts must be consistent:

$$\sum_{i \in N} B_i = \sum_{j \in N} A_j. \tag{2}$$

Three formal rules for separating successive elevator trips from each other were presented in [4]. In general, an elevator trip starts at a stop where passengers board an empty elevator and ends to a stop where the elevator becomes empty again. Hence, at the first node, the predicted boarding count must be at least one and the alighting count zero, and at the last node the reverse must hold:

$$A_1 = 0,\ \ B_1 \geq 1,\ \ A_n \geq 1,\ \ B_n = 0. \tag{3}$$

At every node between the first and the last node, at least one passenger either boards or alights:

$$A_i + B_i \geq 1, \qquad 1 < i < n. \tag{4}$$

By taking into account the assumptions 2 and 3, the constraint in Eq. 4 can be more accurately stated as follows. According to assumption 2, at least one passenger boards at node $r_i \neq n+1$ and alights at node i:

$$r_i \neq n+1 \Leftrightarrow X_{r_i,i} \geq 1, \qquad 1 < i \leq n, \tag{5}$$

and according to assumption 3, passengers cannot alight at a node to which there is no delivery request, and thus, at least one passenger must board:

$$r_i = n+1 \Leftrightarrow A_i = 0 \wedge B_i \geq 1, \qquad 1 < i < n. \tag{6}$$

This condition corresponds to the assumption that the elevator does not stop for nothing. In other words, if the elevator does not stop to serve a delivery request, it must stop to serve a pickup request which corresponds to at least one passenger.

The predicted OD passenger counts are related to the predicted boarding and alighting counts through the flow conservation constraints:

$$\begin{aligned} \sum_{j|(i,j)\in A} X_{ij} &= B_i, \qquad \forall\, i \in N, \\ \sum_{i|(i,j)\in A} X_{ij} &= A_j, \qquad \forall\, j \in N. \end{aligned} \tag{7}$$

The number of passengers in the elevator between the nodes i and $i+1$, P_i, is computed as follows:

$$P_1 = B_1, \qquad P_{n-1} = A_n, \qquad P_i = P_{i-1} + B_i - A_i, \qquad 1 < i < n-1. \tag{8}$$

The elevator capacity is always respected because of the domain of the variables.

The problem of finding the passenger counts for the arcs or OD pairs of an elevator trip such that the predicted boarding and alighting counts are as close as possible to the measured counts can be seen as a network flow problem. In such a problem, the objective function is typically linear. A linear objective function may, however, result in a solution that produces small deviations between most of the predicted and observed counts, but accepts large deviations for some counts. This is not good since the difference between each observed and predicted count should be small. Hence, we consider the LS deviation between the predicted and observed counts as the objective function:

$$\sum_{i\in N}[(A_i - a_i)^2 + (B_i - b_i)^2]. \tag{9}$$

An optimal solution to an ETODM estimation problem is a vector of OD passenger counts X_{ij}, $(i,j) \in A$, that minimizes Eq. (9) with respect to the constraints in Eq. 2-8.

Note that the LS objective value in Eq. 9 is zero only if the problem is consistent. This is the case if:

$$\sum_{i\in N} b_i = \sum_{j\in N} a_j, \tag{10}$$

$$b_i \geq |OD_{ij}|, \qquad \forall\, i \in N, \tag{11}$$

$$a_j \leq \sum_{i|1\leq i<j} (b_i - |OD_{ik}|), \qquad \forall\, j \in N, \tag{12}$$

where $OD_{ij} = \{(i,j) \in N^2 | r_j = i\}$ is the set of OD pairs whose origin node is i, destination node is j and the delivery request to node j is registered at node i. Hence, $|OD_{ij}|$ is the minimum number of passengers that must be assigned from node i to nodes j. If the condition Eq. 11 does not hold, then the assumption 2 is violated. The set $OD_{ik} = \{(i,k) \in N^2 | r_k = i, k \neq j\}$ is the set of OD pairs whose origin node is i, destination node is $k \neq j$ and the delivery request to node k is registered at node i. Hence, the condition in Eq. 12 checks that the total count of passengers that can be assigned to OD pairs ending to node j is equal to or greater than the count of passengers who alight at node j, taking into account the minimum count of passengers that must be assigned to all other destination nodes k of origin nodes i.

Similar consistency conditions were defined in [4] but the corresponding BILS formulation is based on different assumptions. It first uses the observed boarding and alighting counts to divide the nodes to pickup and delivery nodes, and then the delivery requests to define the OD pairs. A node is defined as a pickup node if $b_i \geq 1$, and as a delivery node if $a_i \geq 1$. The disadvantage here is that if, e.g., the observed boarding count is zero even if the true count is positive, then the corresponding node will not be classified as a pickup node. If, in addition, the observed alighting count is zero, then the corresponding stop will not be included in the formulation at all. In both cases, the number of OD pairs will be smaller than it in reality should.

Our formulation is based only on the stops and delivery requests, which means that all stops will always be included in the formulation. Furthermore, according to Eq. 1, node i between the first and the last node defines always an OD pair with all nodes j such that $r_j < i$. This typically increases the number of OD pairs compared to the BILS formulation, which makes our approach more conservative. The two approaches will yield the same set of optimal solutions, if the formulations contain the same set of nodes, and they are consistent. A possible future improvement to the current formulation would be to consider also the variations in elevator load. More specifically, if at any stop i the arrival load is larger (resp. smaller) than the departure load, then $A_i > 0$ (resp. $B_i > 0$) while a constant load during the entire stop suggests that no alighting (resp. boarding) occurred. This would incorporate confidence of the measurements and help to correct unexpected human behavior. Note, however, that even if the arrival load was larger (resp. smaller) than the departure load, it is still possible that B_i (resp. A_i) should be greater than zero. This is because people have different weights. Hence, the load information can be used as an additional source of information but there should be another method to count the boarding and alighting passengers. A further research subject is to study which one of the alternative approaches gives better estimation results. In this paper, we concentrate on studying the effect of randomization and fulfilling real time elevator group control requirements with the proposed approach.

In destination control, passengers use numeric keypads to register destination calls at the elevator lobbies. Each destination call combines a pickup and a delivery request, and if every passenger would always register a destination call, then the OD passenger counts, i.e., ETODMs, would trivially be defined by the number of destination calls. However, it has been observed that people move often in batches and typically only one passenger of the batch registers the call to the destination [23]. It has also been observed that sometimes people abuse the destination control by giving several destination calls. Hence, the destination calls are not in general a reliable way to estimate the ETODMs. They could, however, also be used as an additional source of information.

To illustrate our formulation, consider the following instance: $n = 4$, $C = 20$, $b_1 = 10$, $b_2 = 1$, $b_3 = b_4 = 0$, $a_1 = a_2 = 0$, $a_3 = a_4 = 6$, and $r_1 = r_2 = 5$, $r_3 = r_4 = 1$. Since the condition (10) does not hold, the problem is inconsistent. Fig. 1 shows the corresponding ETODM estimation problem with the predicted OD passenger counts X_{ij}, $i,j \in A$, and the predicted boarding and alighting counts, $B_i, A_i \in [0, C]$, $i = 1,2,3,4$.

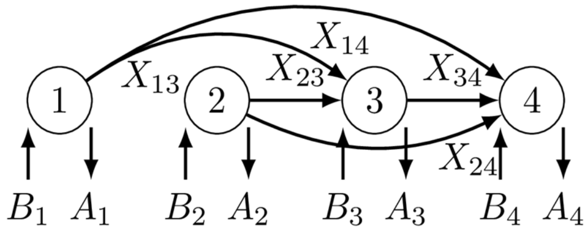

Figure 1 Example ETODM estimation problem

SEARCH ALGORITHMS

We consider a complete standard backtracking search which consists of a depth-first traversal of the search tree. At a node of the search tree, an uninstantiated variable is selected and the node is extended so that the resulting new branches out of the node represent alternative choices that may have to be examined in order to find a solution. The branching strategy determines the next variable to be instantiated, and the order in which the values from its domain are selected.

Branching Strategy and Candidate Algorithms. A branching strategy determines the next variable to be instantiated (variable selection), and the next value the variable is assigned from its current domain (value selection). The branching strategy strongly impacts the performance of the search by improving the detection of solutions (or failures for unsatisfiable problems) when building the search tree.

Here we consider the following variable selection strategies: *dom* (D) selects the variable whose domain is minimal; *dom/wdeg* (W) selects the variable that minimizes the quotient of its domain size over its weighted degree; *lex* (L) selects a variable according to lexicographic ordering; *random* (R) selects a variable randomly [24]. We consider only two classical value selection strategies: *minVal* (M) selects the smallest value and *randVal* (R) selects a value randomly. There is also a third classical value selection strategy, *maxVal*, which selects the largest value. However, our numerical experiments indicated that it is less efficient than *minVal*, and thus, is not considered in this study. A candidate algorithm (CA) is obtained by combining a variable and a value selection strategy. For example, DM uses *dom* for variable selection and *minVal* for value selection.

Optimization Procedure. Most CP tools use by default a standard *top-down* branch-and-bound algorithm which maintains a lower bound, lb, and an upper bound, ub, on the objective value. When $ub \leq lb$, the sub tree can be pruned because it cannot contain a better solution. Here, the problem is solved using the *bottom-up* procedure. The procedure starts with a lower bound, lb, as a target upper bound which is incremented by one unit until the problem becomes feasible. The first solution found by the *bottom-up* procedure is proven optimal. If (by luck) the first solution found by the *top-down* procedure is optimal, the optimality has to be still proven.

Let opt denote the optimal objective value. The *bottom-up* procedure solves $opt - lb$ unsatisfiable problems and only one satisfiable problem before finding an optimal solution. Hence, the number of problems that has to be solved is linear with respect to lb. Most *bottom-up* variants reduce from a linear to a logarithmic number of iterations in the worst-case. The *top-down* procedure is a good candidate if $opt - lb$ is large or the goal is to find good soutions quickly. In our case, the boarding and alighting counts are often measured without errors, and if an error is made, it rarely exceeds one unit. This means that the optimal objective value is often equal or close to zero, and thus, the *bottom-up* procedure with the initial lower bound equal to zero, $lb = 0$, is a good candidate.

NUMERICAL EXPERIMENTS

Simulation Process. We simulated lunch hour traffic in a 25-storey office building using the Building Traffic Simulator (BTS) [25]. The simulation time was 15 minutes. In a typical lunch hour traffic pattern, which was used also in this study, the proportion of incoming, outgoing and inter-floor traffic is 40%, 40% and 20%, respectively. We used a conventional group of eight elevators with the capacity of 21 passengers, and adjusted the traffic intensity so that the handling capacity (HC) of the elevator group was insufficient. When the HC is insufficient, the elevators become often fully loaded, and thus, make many stops during one up or down trip. This increases the number of difficult problem instances. Because, in practice, elevator groups are designed to have enough HC, the problems occurring in reality are likely to be less complex.

Every simulation produces data, e.g., all passengers and their origin and destination floors that are used to construct the true BODM. An element in the true BODM corresponds to the true number of passengers from an origin to a destination. The simulation data are also used to construct the ETODM estimation problem instances. By solving all the ETODM estimation problem instances, and adding up the estimation results, we obtain the estimated BODM.

To obtain several sets of test data, we repeated the simulation 10 times with different seeds. The resulting 10 sets of problem instances contain only consistent instances. To obtain also inconsistent instances, we assumed a measuring accuracy of 90% and created inconsistent problem instances from the consistent problem instances by removing one passenger from each boarding and alighting count with 10% probability. Passengers were removed and not added since experience has shown that, at least with an electronic load weighing device, the observed count is typically one passenger less than the true count, if an error occurs. This resulted in 10 new sets of problem instances containing in total 165 inconsistent instances, which is about 30% of the total of 558 instances in the 10 new sets. This shows that since an elevator trip consists of several stops, the measuring accuracy per stop must be high in order to increase the number of consistent instances which are easier and faster to solve. Although the 10 new sets contain also consistent instances, we call them inconsistent to separate them from the sets containing only consistent instances.

BODM Construction. The BODM of a given time interval or simulation is constructed by adding up the ETODMs estimated during that interval. An ETODM estimation problem may, however, have several optimal solutions. We consider the first 10^K, $K = 0,1,2,3,*$, optimal solutions per instance and select the final solution as the average of the computed solutions. The $*$ sign refers to all optimal solutions. Because of the different branching heuristics, the different CAs will not give the same set of first 10^K optimal solutions, and thus, the final solutions will be different. This means that the BODMs estimated with different CAs will be different except for $K =*$. When we select the final solution to a problem instance as the average of the computed optimal solutions to the instance, we obtain always only one BODM per simulation.

Another reason for selecting the average is that, if only some of the optimal solutions are available, it describes the differences between the CAs with respect to the characteristics of these solutions. Hence, the average makes it possible to compare the CAs with respect to BODM quality. Note that the average of the computed optimal solutions is not in general the same as the continuous solution to the instance.

BODM Quality. The quality of an estimated BODM is evaluated based on the total squared deviation. Let X_{ij}^{true} and X_{ij}^{est} denote the true and the estimated passenger count from origin i to destination j in the true and the estimated BODM, respectively, and let N denote the total number of OD pairs in the building. The total squared deviation is the sum of the OD passenger count deviations between the estimated and the true BODM:

$$\sum_{i\in N}\sum_{j\in N}\left(X_{ij}^{est} - X_{ij}^{true}\right)^2. \tag{13}$$

Hence, the total squared deviation measures the proximity of the estimated BODM to the true BODM with respect to the OD passenger counts.

EXPERIMENTAL RESULTS

All the experiments were conducted on a Linux machine with 32 GB of RAM and a Intel Core i7 processor (6 cores -- 3.20GHz). The implementation is based on *choco* (http://choco.mines-nantes.fr). We consider the deterministic algorithms DM, WM and LM. From the randomized algorithms, we consider only DR and RR for the reasons explained in the following sections.

A randomized search typically gives a different set of optimal solutions per problem instance when it is solved several times. Hence, to study the average performance of the randomized algorithms, we ran DR 50 times for consistent and inconsistent instances, and RR 50 times for consistent instances, but only 5 times for inconsistent instances because of much longer solving times. One run consists of solving all the 558 instances corresponding to the 10 BODMs once, and thus, each run produces 10 estimated BODMs.

Number of Optimal Solutions. Table 1 shows the distribution of the number of optimal solutions among the 558 consistent and inconsistent problem instances. It suggests that the search space is typically larger for the inconsistent instances. Table 2 gives the distribution of the LS objective value at the optimal solutions to the inconsistent problem instances. The distribution shows that the optimum of an inconsistent instance rarely exceeds one, which confirms that the *bottom-up* procedure with the initial lower bound equal to zero, $lb = 0$, is a good choice for optimization.

Table 1 Distribution of the number of optimal solutions

No. sols.	$= 1$	≤ 1	$\leq 10^2$	$\leq 10^3$	$\leq 10^4$	$> 10^4$
Consistent	405	84	35	19	13	2
Inconsistent	320	107	75	27	17	12

Table 2 Distribution of the LS objective value

Objective value	0	1	2	3
Count	393	136	27	2

Solving Time. Let t denote the solving time of a given problem instance for a given value of K, and let t_0 be the minimum solving time among the CAs for the instance and the value of K. Table 3 shows the geometric mean, geometric standard deviation and the maximum of t/t_0 computed over all inconsistent instances and values of K. The geometric mean and standard deviation are used since the arithmetic counterparts are not suitable for normalized values [26]. These results are not shown for consistent instances since the differences between the CAs were negligible. It can be concluded that DM is usually faster and more stable than WM and LM. Hence, *dom* is the best deterministic variable selection strategy with respect to solving time.

Table 3 Performance of the deterministic CAs with respect to solving time

	DM	WM	LM
Geom. mean	1.005	1.036	1.044
Geom. std	1.039	1.050	1.154
Max	2.562	3.710	11.684

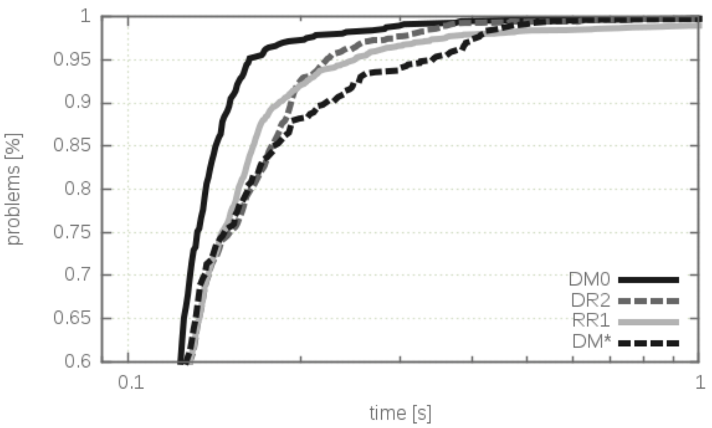

Figure 2 Cumulative distribution of solving time for four selected CAs

Fig. 2 shows the percentage of inconsistent instances solved within a given time for four selected CAs, namely, DM0, DR1, RR2 and DM$*$. The last character corresponds to a given value of K. For example, the graph of DM0 shows that DM can find the first solution to more than 95% of the inconsistent instances in less than 0.2 seconds. Although not shown, all other similar graphs for the deterministic CAs stay within the graphs of DM0 and DM$*$. As shown in the upper right corner of the figure, RR1 takes more time for some instances than DM$*$, which means that randomized variable and value selection is not a good strategy since we can find all solutions with a deterministic algorithm faster. There are, however, a few problem instances to which it takes clearly a longer time to find all solutions as shown by the graph of DM$*$. DR2 produces an acceptable increase in solving time, but although not shown in the figure, the solving times of DR3 become too long. It can be concluded that DR should be preferred over DM for $K = 0,1,2$, if it increases BODM quality.

In [4], the solving time of the BILS algorithm in finding all optimal solutions to four consistent and inconsistent example problem instances is reported. Table 4 shows these results also for DM which is a little bit slower except for the inconsistent instances 3 and 4 for which DM is much faster. In general, the solving time of DM is acceptable considering a real application although for the inconsistent instance 3, the 0.5 seconds limit is somewhat exceeded. The example instances were formulated using the BILS formulation. With the CP formulation, the number of optimal solutions to the inconsistent instances 2, 3, and 4 are 44, 14091 and 155, respectively. This illustrates the differences between the two formulations.

Table 4 Comparison of DM and BILS

	Consistent				Inconsistent			
Instance	1	2	3	4	1	2	3	4
DM solving time [s]	0.13	0.14	0.27	0.2	0.14	0.18	0.66	0.26
BILS solving time [s]	0.00	0.00	0.13	0.08	0.00	0.00	2.24	117.5
No. sols.	1	5	2016	9	5	29	10353	78

Total Squared Deviation. Fig. 3 shows the total squared deviation of the deterministic CAs for inconsistent instances as a histogram. The total squared deviation for each BODM is computed based on Eq. 13, and the results shown in the figure are obtained by summing up these deviations. The CAs are grouped by the parameter K, $K = 0,1,2,3$, and the horizontal line is the total squared deviation for all optimal solutions, which is the same for all CAs. The corresponding histogram for the consistent instances is not shown since it looks exactly the same except that the total squared deviations are smaller. The main result is that finding multiple optimal solutions reduces the deviation. In addition, DM and WM are almost equivalent and LM results always in the greatest deviation, which again makes DR a better choice than WR and LR.

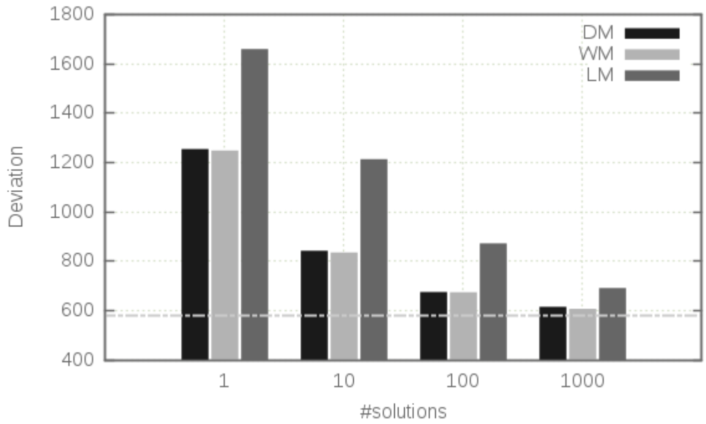

Figure 3 Total squared deviation of the deterministic CAs

Table 5 shows the average total squared deviation of DR and DM for the inconsistent instances. It can be concluded that DR is on average better than DM. However, if we consider the 0.5 seconds limit, then DM is a better choice since based on Fig. 2, DM∗ can solve approximately as many problem instances as DR2 within this limit and the BODM quality of DM3 is already better than that of DR2. For shorter time limits, DR is a better choice.

Table 5 Total squared deviation of DR and DM

K	0	1	2	3
DR	1179.0	762.1	653.4	599.2
DM	1255.0	843.5	676.9	616.5

Number of Passengers. For inconsistent instances, the total number of passengers in the estimated BODM is typically less than in the true BODM. The reason for the underestimation is naturally that the inconsistent instances were created by removing passengers from the true counts. However, underestimation is an issue also in reality and, as shown in Table 5, the amount of underestimation depends on the CA. The amount of underestimation is obtained by subtracting the total number of passengers in the true BODM from the total number of passengers in the estimated BODM for each of the 10 BODMs, and then adding up these differences. Note that also overestimation might be an issue in practice but, as with underestimation, this depends on the measuring accuracy and the used measuring device and method.

Table 6 Underestimation of the number of passengers

	DM	DR				RR			
K		Avg	Std	Min	Max	Avg	Std	Min	Max
0	175.0	16.2	6.0	10.0	30.0	22.6	8.5	14.0	33.0
1	79.3	35.1	1.6	32.1	38.8	38.4	2.0	36.1	40.9
2	58.7	39.1	0.6	38.0	40.9	40.9	0.9	39.8	42.1
3	50.6	40.6	0.3	40.1	41.3	42.0	0.5	41.5	42.6

Table 6 shows that both randomized CAs significantly reduce the underestimation, especially when only one optimal solution per instance is computed. Even the worst cases (max) are better than the results obtained with DM. DR results on average in smaller underestimation than RR. Furthermore, the more optimal solutions are computed the more accurate and the more stable is the estimated BODM. These results support the selection of DR over DM.

CONCLUSIONS

We presented a constraint programming (CP) formulation for the elevator trip origin-destination matrix (ETODM) estimation problem. An elevator trip consists of successive stops in one direction of travel with passengers inside the elevator, and the estimated OD matrix contains the OD passenger counts for the OD pairs of the trip. The ETODMs estimated for a given time interval are added up to construct the building OD matrix (BODM) of that interval. The passenger traffic in a building can be learned by combining the BODMS of the same day or time interval, and usually day of week. These matrices can be used to make forecasts about future passengers. The forecasts are needed in elevator dispatching to improve dispatching decisions with respect to future passengers.

An ETODM estimation problem may have many optimal solutions, and any of these solutions may correspond to what happened in reality. To obtain robust forecasts, the learned BODMs should describe the possible realizations of the passenger traffic as well as possible. This can be achieved by finding all or several optimal solutions to each problem instance and selecting the final solution, e.g., randomly or as the average of the optimal solutions.

We compared three deterministic and two randomized CP algorithms in finding a predefined number of optimal solutions to the ETODM estimation problem. Several test problems were obtained by simulations of lunch hour traffic in a typical multi-storey office building. The traffic intensity was adjusted above the handling capacity of the simulated elevator group. This resulted in complex problem instances that enable robust performance testing and comparison of the algorithms.

The comparison of the algorithms was based on solving time and BODM quality which affects the reliability of the passenger traffic forecasts. The results suggest that randomization and multiple optimal solutions is a good compromise between solving time and quality. For very complex problem instances, the fastest CP algorithm turned out to be even faster than the previous estimation approaches and algorithms. In addition, the proposed approach fulfils real time elevator group control requirements for solving ETODM estimation problems.

REFERENCES

[1] T. Tyni and J. Ylinen, “Genetic algorithms in elevator car routing problem”. *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO-2001)*, 1413-1422 (2001).

[2] J. Koehler and D. Ottiger, “An AI-based approach to destination control in elevators”. *AI Magazine*, Vol. 23, No. 3, 59-78 (2002).

[3] P.B. Luh, B. Xiong and S-C. Chang, “Group elevator scheduling with advance information for normal and emergency modes”. *IEEE Transactions on Automation Science and Engineering*, Vol. 5, No. 2, 245-258 (2008).

[4] J-M. Kuusinen, J. Sorsa and M-L. Siikonen, “The elevator trip origin-destination matrix estimation problem”. *Transportation Science,* ePub ahead of print May 8, http://pubsonline.informs.org/doi/abs/10.1287/trsc.2013.0509 (2014).

[5] M-L. Siikonen, “Elevator group control with artificial intelligence”. *Research Report A67,* Systems Analysis Laboratory, Helsinki University of Technology (1997).

[6] G.R. Strakosch, *Vertical Transportation: Elevators and Escalators*. Jon Wiley & Sons, Hoboken, New Jersey (1983).

[7] S. Nguyen, “Estimating origin-destination matrices from observed flows”. *Transportation Planning Models,* 363-380. Elsevier B.V., Amsterdam (1984).

[8] H. van Zuylen and L. Willumsen, “The most likely trip matrix estimated from traffic counts”. *Transportation Research B*, Vol. 14, No. 3, 281-293 (1980).

[9] B. Lamond and N. Stewart, “Bregman’s balancing method”. *Transportation Research B*, Vol. 15, No. 4, 239-248 (1981).

[10] M. Bell, “The estimation of an origin-destination matrix from traffic counts”. *Transportation Science*, Vol. 17, No. 2, 198-217 (1983).

[11] M. Ben-Akiva, P. Macke and P. Hsu, "Alternative methods to estimate route-level trip tables and expand on-board surveys". *Transportation Research Record*, No. 1037, 1-11 (1985).

[12] S. Tsygalnitsky, "Simplified methods for transportation planning". *Master's thesis*, Department of Civil Engineering, Massachusetts Insitute of Technology, Cambridge (1977).

[13] Y. Li and M. Cassidy, "A generalized and efficient algorithm for estimating transit route ODs from passenger counts". *Transportation Research B*, Vol. 41, No. 1, 114-125 (2007).

[14] B. Li, "Markov models for Bayesian analysis about transit route origin-destination matrices". *Transportation Research B*, Vol. 43, No. 3, 301-310 (2009).

[15] H. Spiess, "A maximum likelihood model for estimating origin-destination matrices". *Transportation Research B*, Vol. 21, No. 5, 395-412 (1987).

[16] E. Cascetta and S. Nguyen, "A unified framework for estimating or updating origin/destination matrices from traffic counts". *Transportation Research B*, Vol. 22, No.6 , 437-455 (1988).

[17] M. Maher, "Inferences on trip matrices from observations on link volumes: a Bayesian statistical approach". *Transportation Research B*, Vol. 17, No. 6, 435-447 (1983).

[18] E. Cascetta, "Estimation of trip matrices from traffic counts and survey data: a generalized least squares estimator". *Transportation Research B*, Vol. 18, No. 4-5, 289-299 (1984).

[19] M. Bell, "The estimation of origin-destination matrices by constrained generalized least squares". *Transportation Research B*, Vol. 25, No. 1, 13-22 (1991).

[20] K. Yoneda, "Elevator trip distribution for inconsistent passenger input-output data". *Decision Making in Manufacturing and Services*, Vol. 1, No. 1-2, 175-190 (2007).

[21] J-M. Kuusinen, M. Ruokokoski, J. Sorsa and M-L. Siikonen, "Linear programming formulation for the elevator trip origin-destination matrix estimation problem". *Proceedings of the 2nd International Conference on Operations Research and Enterprise Systems*, 298-303 (2013).

[22] F. Rossi, P. van Beek and T. Walsh, *Handbook of Constraint Programming (Foundations of Artificial Intelligence)*. Elsevier Science Inc., New York (2006).

[23] J-M. Kuusinen, J. Sorsa, M-L. Siikonen and H. Ehtamo, "A study on the arrival process of lift passengers in a multi-storey office building". *Building Services Engineering Research & Technology*, Vol. 33, No. 4, 437-449 (2012).

[24] F. Boussemart, F. Hemery, C. Lecoutre and L. Sais, "Boosting systematic search by weighing constraints". *Proceedings of the 16th European Conference on Artificial Intelligence, ECAI'2004*, 146-150 (2004).

[25] M-L. Siikonen, T. Susi and H. Hakonen, "Passenger traffic flow simulation in tall buildings". *Elevator World*, Vol. 49, 117-123 (2001).

[26] P. Fleming and J. Wallace "How not to lie with statistics: the correct way to summarize benchmark results". *Communications of the ACM*, Vol. 29, No. 3, 218-221 (1986).

BIOGRAPHY

Juha-Matti Kuusinen received MSc in applied mathematics from Helsinki University of Technology, Finland, in 2009. He currently works in KONE Technology department and is finalizing the Doctor of Technology degree in the Systems Analysis Laboratory, Aalto University, Finland, where he has also been a visiting lecturer.

Dr Arnaud Malapert is an associate professor at the University of Nice Sophia Antipolis. He received a Master's Degree in 2006 in Computer Science from the University Pierre and Marie

Curie (Paris VI). He received a PhD in 2011 in Computer Science from the University of Nantes entitled "Shop and batch scheduling with constraints". His research interests are mainly related to constraint programming and operations research.