The heart of your building™

PUBLICATION

OPTIMAL CONTROL OF DOUBLE-DECK ELEVATOR GROUP USING GENETIC ALGORITHM

Janne Sorsa
E-mail: janne.sorsa@kone.com
Marja-Liisa Siikonen
E-mail: marja.liisa@siikonen@kone.com
Harri Ehtamo

This paper was presented at IFORS 2002, 16th Triennial Conference of International Federation of Operational Research Societies and first published by Blackwell Publishing in International Transactions in Operational Research. The definitive version is available at www.blackwell-synergy.com.

Full bibliographic reference:
J. Sorsa, M-L. Siikonen, H. Ehtamo. Optimal control of double-deck elevator group using genetic algorithm. International Transactions in Operational Research, Vol. 10 (2003), pp. 103-114.

ABSTRACT

We shall introduce the principles of optimal routing of double-deck elevators. Elevator routing problem is formulated as an integer programming problem and it is solved using a genetic algorithm in a real time system. The optimal routes of double-deck elevators have not been considered earlier in the literature. The simulation results are analysed with the discussion about the significance of the method.

Keywords: Elevator control, vehicle routing, optimisation, simulation

1 INTRODUCTION

Mathematical methods to control an elevator group have been studied quite frequently since Closs (1970) applied dynamic programming to elevator routing. Even if the topic has been researched widely, no real systems using optimisation were developed until Tyni and Ylinen (1999a, 2001). In their method passenger landing calls are allocated to single-deck elevators in such a way that the routes of elevators are optimal. In this paper, the optimal routes of double-deck elevators are solved in real time using genetic algorithm.

The control of a double-deck elevator group has not been researched much in literature. The existing double-deck elevator systems use relatively simple methods to determine the deck, which serves the landing calls (e.g. Nowak and Luce, 1986). An advanced method, KONE BestDeck, was described in Siikonen (2001). Ylinen and Tyni (2001) patented a genetic procedure to allocate landing calls to multi-deck elevators.

In high-rise buildings, groups of elevators serve the passenger traffic. In slender buildings where all elevators leave from the main lobby, the elevator groups can take over 50% of the building core space. In double-deck elevators, two cars are attached to each other so that they serve sequential floors simultaneously. The handling capacity per shaft is 1.5–2.0 times the capacity of a single-deck elevator. The typical core area reduction with double-deck elevator groups is 25–40% compared to single-deck elevator groups.

Currently in Europe lots of buildings with 20–40 floors are being built. In these buildings, zoned single-deck elevator groups are normally used, but to save building core space double-deck elevator systems offer an attractive alternative. In some cases, only one double-deck elevator group instead of two single-deck groups serving different zones can serve the whole building.

We study an optimisation model for elevator routing. The model captures the special constraints of elevator systems. Elevator routing has not been considered previously as an integer programming problem. The model can be generalised to optimise routes of an elevator group.

We develop a real time method using genetic algorithm to solve the optimisation problem. The method is the first application of optimal control policies to double-deck elevator group and it can be applied to real elevator products immediately. In this paper, only double-deck elevators are considered, but the method can be generalised easily to control multi-deck elevator groups as well. The method is also capable of handling hybrid elevator groups, where each elevator can contain an arbitrary number of cars.

An example building with 29 floors is simulated with a double-deck elevator group consisting of eight elevators. Two single-deck elevator groups of six elevators are needed to serve this building. Using double-deck elevators instead of single-deck elevators saves about 33% of the shaft space in this case. The travelling convenience of the passengers is increased because the double-deck elevator group serves all floors of the building, whereas the single-deck groups serve separate zones of the building.

We use three methods to control the elevator group: BestDeck (Siikonen, 2001) and the optimal control method with two different objective functions. The simulation results show that the optimal control provides the best service level. Although none of the methods was the best in all traffic situations and with all key figures, the objective functions in optimal control can be fine-tuned more easily than BestDeck to obtain better results.

In Section 2, some historical background to elevator systems is provided. The methods applied to elevator group control during the last decades are described. An optimisation model for an elevator group is discussed in Section 3 with emphasis on the special constraints of elevator systems. The solution method to the problem is based on the genetic algorithm depicted in Section 4, and the simulation results are presented in Section 5. The applicability of the method in future elevator products and the relevance of the optimisation model are discussed in Section 6.

2 ELEVATOR SYSTEMS

2.1 Double-deck elevators in high rise buildings

Building core space can be saved, if more than one elevator is put in the same elevator shaft. The biggest savings in core space are obtained with shuttle elevators that serve the traffic between main lobby and the sky lobby. This kind of an arrangement was first built in Sears Tower in Chicago in 1960s. Also many other tall buildings and double deck elevator groups serving all floors were built at that time. Elevator group controls were then made with relay technology and the control logic was quite simple. Passenger journey times became long and double deck systems were not considered very attractive alternative to single deck elevators then.

The 1990s saw another boom to build tall buildings, especially in the Far East. The technology had improved a lot; elevators move smoothly with heavier loads using smaller machines and computers to control the elevators. With computers, more complex logic and mathematical methods in group control systems can be applied, which makes them very efficient. For example, the elevator system of Petronas Tower, the tallest building in the world at the time of writing, contains several double-deck elevator groups.

2.2 Group control principles

Within an elevator group, the elevators serve the same calls given from the landing floors. By sharing the landing calls between several elevators, the efficiency of the service is improved and the waiting times of passengers become shorter. The landing calls are allocated to elevators either continuously or immediately. In a continuous call allocation system, all active landing calls are allocated to elevators twice a second. Another approach is immediate call allocation, where the calls are allocated immediately after the passenger has pressed the landing call button. Both systems have their own advantages and disadvantages: continuous allocation is more efficient, but immediate allocation is more attractive when considering the psychological aspect of waiting times.

There have been several efforts to solve the call allocation problem. As the earliest group controllers were based on relay technology it was not possible to optimise the call allocation. Interconnected Full Collective (IFC) control was the first significant call allocation method for elevator groups with more than one elevator. Passengers give calls from their landing floors, either up or down. Group controller recognises the floor and the direction of the call. The nearest elevator, which is not full of passengers, travelling in the direction of the given call is allocated. The serving elevator picks up the passengers behind landing calls in its travelling direction and serves the given car calls in correct order.

The principle of collective control is a near-optimal solution to routing of single elevator (Closs, 1970) and it is adapted widely in subsequent group control methods. However, for larger elevator groups, the collective control resulted in long waiting times especially in the lower part of the building. The inefficiency of collective control resulted from the basic philosophy of the method as the landing calls received higher priority only after being active for certain time period. Another consequence of the method was bunching of the elevators during heavy outgoing traffic. The Enhanced Spacing Principle (ESP), which was developed in KONE Corporation during the 1980s, solved the problem by forecasting the landing call times. Calls with long waiting-time forecasts are served before the actual waiting time becomes long. The change of philosophy resulted in a significant reduction in waiting times.

In the 1960s, Otis Elevator Company applied the collective control principle to double-deck elevators with relay technology (patented later by Nowak and Luce, 1986). The method, so called “Trailing Deck”, allocates landing calls to the trailing deck of the elevator with respect to the travelling direction, as the name of the method suggests. The leading deck serves only coincident calls, i.e. landing and car calls adjacent to the allocated landing in the travelling direction of the elevator. This method has the same disadvantage of grouping as the collective control for single-deck elevators. One side effect of the method is that usually the trailing deck has significantly greater load compared

Janne Sorsa & Marja-Liisa Siikonen & Harri Ehtamo

to the leading deck, which leads to a significant loss of handling capacity, especially in inter-floor traffic.

KONE BestDeck (Siikonen, 2001) is a method for allocating the landing call to a specific deck of a previously selected elevator. BestDeck uses a heuristic criterion to select the deck, for example the carload, which results in more efficient use of the capacity of a double-deck elevator. BestDeck offers also a benchmark in service level for optimal control of a double-deck elevator group.

2.3 Mathematical methods used in group control

Artificial intelligence and expert systems are used in most of current group control systems. The Japanese elevator companies, in particular, have adapted fuzzy group controllers using neural networks to learn the traffic patterns of the building. Imasaki et al. (1995) described a system in which the traffic situation is recognised by a neural network and the elevators are allocated according to fuzzy rules. A similar system was investigated in Markon et al. (1994) and Sasaki et al. (1996). In the group controller of KONE Corporation, the traffic patterns are learned and recognised by fuzzy rules (Leppälä, 1991).

In Closs (1970), the principle of optimal elevator routing was defined from the perspective of dynamic programming. The optimisation was found highly constrained because of passengers' expectations of elevator behaviour. For example, the elevator is not allowed to reverse its direction when it is transporting passengers to their destinations. As Closs (1970) concluded, "it is difficult to represent the constraints mathematically because these constraints are applied to the way in which states may be reached rather than to specific states".

Tyni and Ylinen (1999a, 2001) developed a method based on genetic algorithm to optimise elevator routing. The method decides the optimal call allocation by minimising the average landing call time. The algorithm converges to the optimum in real time and is one of the first applications of a genetic algorithm in control systems.

3 OPTIMISATION MODEL FOR AN ELEVATOR GROUP

3.1 Objective functions

The call allocation problem can be considered a problem of defining the optimal routes of elevators. In optimisation different cost functions can be used. A default method is to minimise total call time of elevators to active landing calls. When minimising call time or other time-related measurement, the estimated times of arrivals (ETA) to stops along the route are used as basic building blocks in objective functions.

$$ETA_s^{l,d} = t_{drive} + S_l t_{stop} \tag{1}$$

Estimated time of arrival to stop *s* (of elevator *l* and deck *d*) can be calculated by using equation (1) above. Drive time of the elevator from current floor to stop floor is denoted by *tdrive*, which is calculated on the basis of kinematics of the elevator. The number of stops (*Sl*) the elevator is required to serve before stop s, is calculated from the route of the elevator. The total number of stops includes the effect of coincident stops, which is shown in detail in Sorsa (2002). The constant stopping time of the elevator is denoted by *tstop*.

Traditionally the call time has been used to measure the performance in elevator system design and analysis. Call time is the only service parameter that can be measured in real elevator system.

$$CT_{total} = \sum_{l=1}^{L} \sum_{d=1}^{D} \sum_{s=1}^{S_{l,d}} ETA_s^{l,d} \tag{2}$$

The objective function to minimise total call time of the whole elevator group is shown in the former equation. The equation is simply the sum of estimated times of arrivals to stops over all decks and elevators. Only stops, where a landing call is active, are included in equation (2). In other words, the service time of passengers inside elevators are not taken into account.

Passenger journey time puts more emphasis on the total time the passenger has to spend in elevator system. One way to calculate the total journey time of all passengers in the elevator system is shown in equation (3) below. It is seen immediately that the objective function for journey times is significantly more complex than the objective function for call times.

$$JT_{total} = \sum_{l=1}^{L} \sum_{d=1}^{D} \sum_{s=1}^{S_{l,d}} \left\{ p_w \left(ETA_s^{l,d} + CWT_s \right) + ETA_s^{l,d} \frac{p_c}{n_c} \right\} \tag{3}$$

Janne Sorsa & Marja-Liisa Siikonen & Harri Ehtamo

Equation (3) divides logically into two parts. First part describes the waiting time of passengers behind landing calls. There are few methods to calculate the expected number of waiting passengers (*pw*). It can be forecast from traffic statistics or from the call waiting time (*CWTs*) and traffic intensity. The second part of the equation (3) determines the time remaining for the rest of the passengers' journey. The expected number of passengers leaving at the stop is calculated by dividing the total number of passengers inside the deck (*pc*) with total number of car calls (*nc*).

The difficulty of calculating the journey time of passengers can be seen in equation (8) immediately. The number of passengers to enter or leave the elevator cannot be calculated exactly. The expected number of passengers behind a landing call depends on the traffic type (up or down peak or mixed) and its intensity. Neither is the origin floor of the passenger known, so it is impossible to calculate the time passenger has already spent in the elevator. Regardless of these difficulties, the control of an elevator group can be optimised even if the probabilistic estimations incur some bias to the objective function.

3.2 Model constraints

The elevator routing problem is similar to the *Travelling Salesman Problem* (TSP), where an elevator visits floors of a building. The routing of an elevator is easily generalised to routing an elevator group as a *Multiple Travelling Salesmen Problem* (MTSP). However, the formulation of TSP is not fully capable of representing the special constraints of an elevator system recognised by Closs (1970).

In Pickup and Delivery Problem (PDP) the route of the serving vehicle is defined between two special nodes in the transportation network, the starting and end-points, which are represented by +0 and –0 respectively. The route consists of customer pickup nodes, +*i*, and delivery nodes, -*i*, where $i \in N$ and *N* is the set of customers. An example of a PDP-graph adapted from Ruland (1995) is presented in Figure 1.

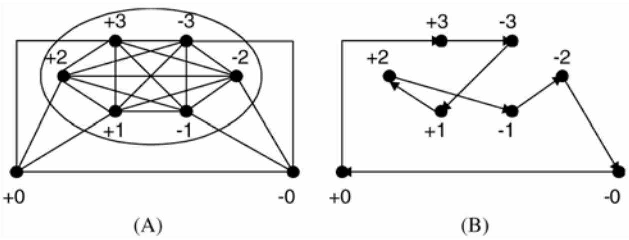

Figure 1. Example of PDP-graph (A) and a feasible PDP-route (B)

The graph defined by pickup and delivery nodes can be considered a normal TSP-graph. The requirement of partial ordering in a feasible route is the principal difference between PDP and TSP.

Janne Sorsa & Marja-Liisa Siikonen & Harri Ehtamo

$$+0 \preceq +i \qquad \forall i \in N \tag{4}$$

$$-i \preceq -0 \qquad \forall i \in N \tag{5}$$

$$+i \preceq -i \qquad \forall i \in N \tag{6}$$

In the former equations the partial ordering is formalised using notation from order theory. All pickup and delivery nodes must be between starting and end points (equations 4 and 5). All pickup nodes must precede the corresponding delivery node (6). A characteristic vector describes a feasible PDP-route. Components can obtain either the value 1 if a certain edge of the graph, for example from –1 to +3 in Figure 1, is included in the route, or the value 0 otherwise. (Ruland, 1995)

The construction of the complete integer programming formulation of the PDP was shown in Ruland (1995) and it is not considered here any further. The special requirements of an elevator routing problem are specified using above notation. Complete integer programme of the problem can be formulated with the following additions.

In the elevator routing problem, the customer pickup and delivery nodes correspond to a landing and car call, respectively. Usually there are several passengers behind one landing call travelling to different destinations. The current position of an elevator is used as a starting point of the route. The end point represents the reversal floor of the elevator, which can be defined to be either the last car call floor or the last floor in the travelling direction of the elevator.

The direction of the elevator plays a crucial part when allocating landing calls; thus in the complete model separate PDP-graphs have to be included for both directions. The precedence of the separate graphs has to be satisfied so that the elevator does not change its direction too early. To force a correct order of serving the directions, the ending point of the first graph must precede the starting point of the second graph, as shown in the equation (7) below.

$$(-0)_1 \preceq (+0)_2 \tag{7}$$

The model is first developed for an elevator travelling upwards. Similar constraints can then be applied to an elevator travelling downwards by changing the floor comparisons.

$$
\begin{aligned}
&+k \preceq +i, \text{if } f_{+k} < f_{+i} \le f_{\max}, \quad && k,i \in N \\
&-k \preceq -i, \text{if } f_{-k} < f_{-i} \le f_{\max}, \quad && k,i \in N \\
&+k \preceq -i, \text{if } f_{+k} < f_{-i} \le f_{\max}, \quad && k,i \in N \\
&-k \preceq +i, \text{if } f_{-k} < f_{+i} \le f_{\max}, \quad && k,i \in N
\end{aligned} \tag{8}
$$

The principle of collective control is included in the model by applying the former precedence constraints, where f denotes the floor of the building. The constraints specify that a service request (landing call or car call) k at a floor lower than the service request i must precede i. The difficulty to model elevator routing is also caught by these equations since the order of the service is not determined from system states (nodes in the PDP-graph) directly.

So far the route of only one elevator has been considered. Real group controller can handle up to eight elevators, which is the situation where optimisation is really needed. However, the developed principles can be generalised to a multi-deck elevator group as well. All decks of elevators of the group shall have its own PDP-graphs to represent its route. Active landing calls in the elevator group are added to all PDP-graphs as customer pickup points. Active car calls are added to the PDP-graph of the deck, in which the car call has been given. The only additional complexity of a multi-deck elevator is reflected in the constraints of changing the direction: end points in one direction of a certain elevator must precede the starting points in the other direction.

In reality the optimisation of group controller should be extended to cover more than one round trip of elevators, which can be achieved by adding required amount of PDP-graphs to the model. “Normal” optimisation constraints, such as the capacity of an elevator, have not been considered so far. The inclusion of such constraints in the model is a straightforward process after the difficulties of elevator routing have been overcome.

4 SOLUTION BY GENETIC ALGORITHM

Solution to the optimisation model can be obtained in real time by specially designed call allocation algorithms. The call allocation methods for the multi-deck elevator group introduced in the following sections are based on the call allocation algorithm for a single-deck elevator group developed by Tyni and Ylinen (1999a). The algorithm avoids the disadvantages of collective control and improves the performance of an elevator group compared to other control methods.

The call allocation algorithm consists of two separate parts. The elevator model handles all the available information from the group controller such as landing calls, and elevator status such as position, car calls and carloads. The genetic algorithm produces solution proposals independently from the search space defined by the elevator model. The solution proposal defines the serving decks for each landing call and new directions for vacant elevators. The elevator model interprets the allocation proposals and calculates the fitness of the proposal. After the last generation in the genetic algorithm has been run, the proposal with the smallest fitness value is chosen as the final allocation decision and is transmitted to the elevators.

The search space for the genetic algorithm is created from the active landing calls and moving states of the elevators in the beginning of each call allocation cycle (see Figure 2). The search space defines the values the genetic algorithm is allowed to use in solution proposals. The solution proposal, or a chromosome in the genetic algorithm, consists of call genes and direction genes. There exists one call gene per active landing call. The value range of a call gene consists of all decks of the group allowed to serve the landing call. A direction gene exists for only those elevators that are standing vacant without a travelling direction. Direction genes can obtain values ‘up’ and ‘down’ only.

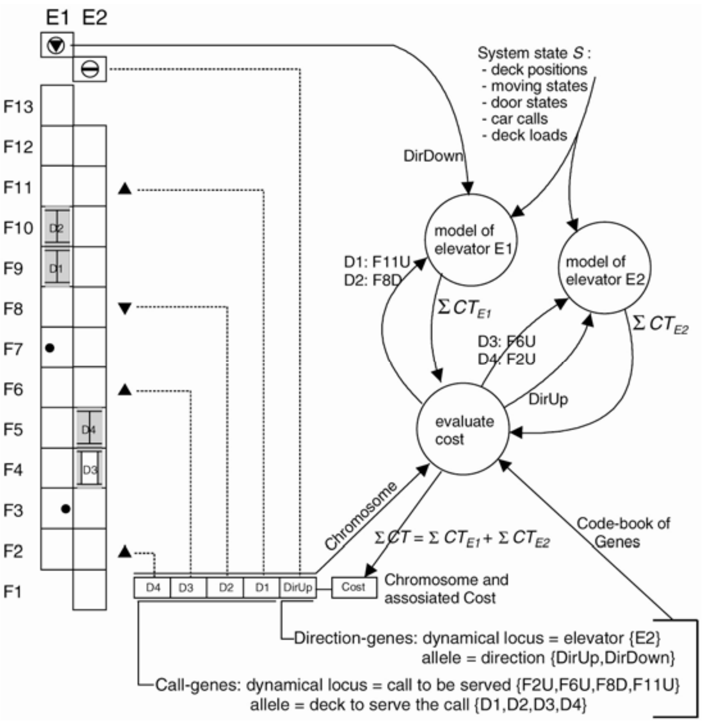

Figure 2. Construction of search space and interpretation of solution proposal

Janne Sorsa & Marja-Liisa Siikonen & Harri Ehtamo

$$n_{genes}^{\max} = 2(N-1) + L \tag{9}$$

$$n_{solutions}^{\max} = (D \cdot L)^{2(n-1)} \cdot 2^{L} \tag{10}$$

The maximum number of genes in a chromosome is given by (9) for an elevator group with n served floors and L elevators. The maximum number of solution alternatives is given by (10), in which D stands for the number of decks in the elevators. All possible landing calls are not active simultaneously, so the upper bounds for the size of the optimisation problem are slightly overstated. However, simulations have revealed that about 75% of possible landing calls might be active during intense lunch-hour traffic.

Decks serving landing calls are extracted from the solution proposal. The route of each deck is constructed independently of the other decks first. The route consists of stops, which are due to either active car calls of the deck or landing calls allocated to the deck according to the solution proposal. The complete routes of elevators are constructed from the individual routes of the decks in such a way that all stops are included in the correct order. The estimated times of arrivals to the stops along the route of the elevator are calculated. Coincident stops, i.e. stops where more than one deck loads or unloads passengers, are recognised and given the same estimated time of arrival. Finally, the cost function is applied to different routes and the fitness of the solution proposal is determined.

Real-time execution of the allocation algorithm is achieved by applying GeneBank technology (Tyni and Ylinen, 1999). Evaluating the cost function consumes most of the computation time required by the call allocation algorithm, so the fitness values of the solution proposals are stored in GeneBank. If the genetic algorithm creates an identical solution at a later stage of the iteration, the fitness of the solution can be retrieved immediately from GeneBank without heavy computations. During the final iterations of the genetic algorithm, GeneBank affects mostly the speed of the algorithm as the populations of the genetic algorithm consist of only a few differing chromosomes. Tyni and Ylinen (1999) found that the computation time of a call allocation algorithm was reduced by 65% with GeneBank.

BestDeck-method has been implemented as a by-product of the researched optimal control method for a double-deck elevator group. The genetic allocation algorithm is “cheated” to handle a multi-deck elevator as one huge single-deck elevator. The genetic algorithm determines the elevator to serve the landing call but does not recognise coincident stops. The deck of the elevator, which will finally serve the landing call, is selected by heuristic criterion. In this implementation, a less crowded deck is allocated. Where there

Janne Sorsa & Marja-Liisa Siikonen & Harri Ehtamo

is an equal carload, the lower deck is allocated. If there is a coincident landing call, it is allocated to the other deck.

5 NUMERICAL SIMULATION

A building with 29 populated floors was simulated during lunch-hour and down-peak traffic. Lunch-hour traffic is the most difficult situation for elevator handling capacity. The biggest differences in the efficiency of group control methods can be observed during the down peak. The elevator group consisted of eight double-deck elevators. The maximum speed of the elevators was 5 m/s and one deck had the capacity of 21 passengers. Two single-deck elevator groups consisting of six elevators are needed to serve the building. Both optimal control and BestDeck methods were simulated. The average call and journey times' minimising objective functions were used as optimisation criteria in optimal control.

The average waiting times and journey times from simulations during lunch-hour traffic are presented in Figures 3 and 4. The handling capacity of the group is exceeded when average carload approaches 80%. The numbers beside the curves show the average carload factor. Simulation results are shown for traffic intensity values below the handling capacity. The call allocation methods provide almost an equal service level during lunch-hour traffic. With traffic intensities less than 13%, the method, which minimises average journey time, results in a significantly longer average waiting time compared to other methods. However, with very intense traffic, the minimisation of average journey times increases the handling capacity. The journey time optimisation results in 10–15% shorter average journey times compared to other methods. The average waiting time should not exceed 30 seconds. The elevator group can handle acceptably about 12.5% traffic intensity with journey time optimisation and about 15% traffic intensity with BestDeck and call time optimisation.

Janne Sorsa & Marja-Liisa Siikonen & Harri Ehtamo

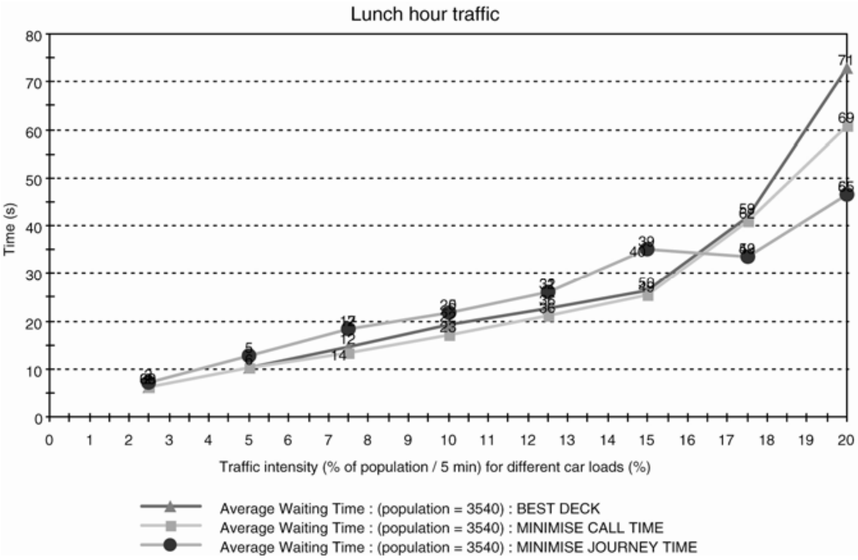

Figure 3. Average waiting times in lunch-hour traffic.

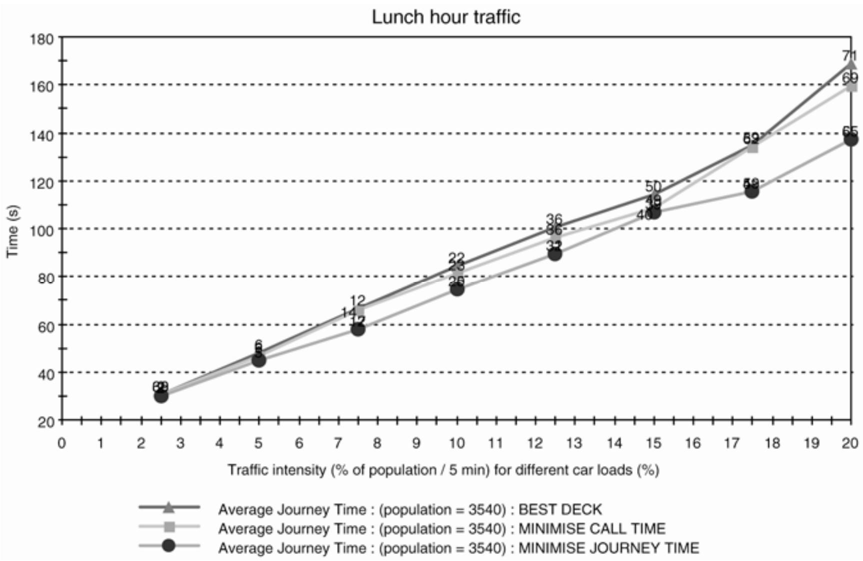

Figure 4. Average journey times in lunch-hour traffic.

During down-peak traffic, significantly shorter average journey times (Figure 6) are achieved when minimising passenger journey times compared to other methods. BestDeck and landing call time minimisation behave similarly, according to journey times. However, when using the BestDeck-method, the average waiting times (Figure 5) increase fast with traffic intensities greater than 20% of population in five minutes. In this light, BestDeck is the worst method of handling down-peak traffic. Average waiting times have a similar pattern during down-peak and lunch-hour traffic, but the superiority of one method over the other is more distinguishable.

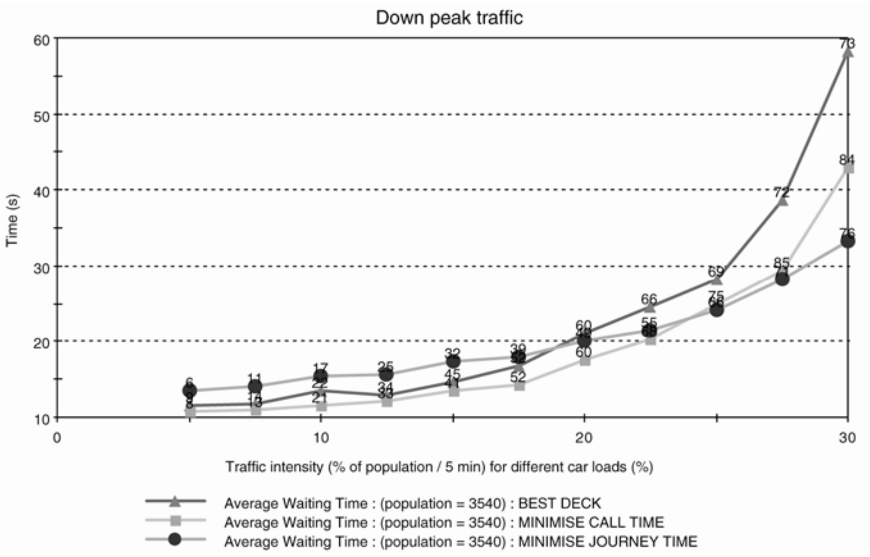

Figure 5. Average waiting times in down peak.

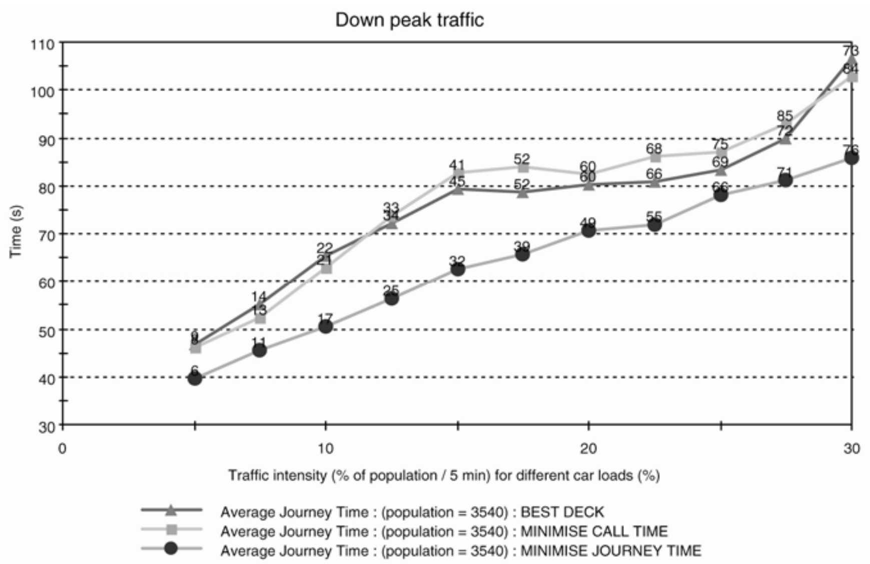

Figure 6. Average journey times in down peak.

6 CONCLUSION

The call allocation methods described in this paper have the potential to be used in real control systems. The optimisation model proposed can be easily generalised to elevators with more than two decks. In practice, there will probably not be multi-deck elevators for a decade. Multi-deck elevators will be used as shuttle elevators mostly, because they are not flexible enough to serve as a local elevator group as a double-deck elevator is. A shuttle group consisting of triple-deck elevators would be efficient and save plenty of core space. In emergency situations, when the whole building has to be refilled quickly, such efficiency is required.

All the presented methods to control a double-deck elevator group can be implemented in real elevator products. Even though the optimal control provides the best service level, BestDeck performed surprisingly well. The best objective function used in optimisation cannot be determined. Rather, the best strategy might be to use landing call time minimisation during normal traffic and passenger journey time minimisation during intense traffic. Another approach would be to design an objective function that is a compromise between landing call and journey time. However, the desirable behaviour of double-deck elevators should be defined in terms of performance and service level. The implementation of the objective function should be relatively simple, and probably should be defined in each building separately, according to the needs of the building usage.

REFERENCES

Closs, G.D., 1970. The computer control of passenger traffic in large lift systems. Ph.D. Thesis, The Victoria University of Manchester. 106 pages.

Imasaki, N., Kubo, S., Nakai, S., Yoshitsugu, T., Kiji, J., Endo, T., 1995. Elevator Group Control System Tuned by a Fuzzy Neural Network Applied Method. In: Proceedings of 1995 IEEE International Conference on Fuzzy Systems, Yokohama, Japan, 1995.

Leppälä, J., 1991. Forecasting passenger traffic in group control of elevators, Helsinki University of Technology, Master of Science Thesis. (In Finnish)

Markon, S., Kita, H., Nishikawa, Y., 1994. Adaptive Optimal Elevator Group Control by Use of Neural Networks. Transactions of the Institute of Systems, Control and Information Engineers. Vol. 7, No. 12, 487-497.

Nowak, F.H., Luce, J.C., 1986. Multicompartment Elevator Call Assigning, Otis Elevator Company, U.S. Patent No. 4 632 224. December 20,1986.

Ruland, K.S., 1995. Polyhedral solution to the pickup and delivery problem. Dissertation, Washington University, Sever Institute of Technology, Department of Systems Sciences and Mathematics. 204 pages.

Sasaki, K., Markon, S., Nakagawa, M., 1996. Elevator group supervisory control system using neural networks. Elevator World. Vol. 44, No. 2, 80-86.

Siikonen, M-L., 2001. Procedure for Control of an Elevator Group Consisting of Double-Deck Elevators, which Optimizes Passenger Journey Time, KONE Corporation, U.S. Patent No. 6 237 721. May 29, 2001.

Sorsa, J., 2002. Optimal group control of double-deck elevators, Helsinki University of Technology, Master of Science Thesis. (In Finnish)

Tyni, T., Ylinen, J., 1999a. Method and apparatus for allocating landing calls in an elevator group, KONE Corporation, U.S. Patent No. 5 932 852. August 3, 1999.

Tyni, T., Ylinen, J., 1999b. Improving the Performance of Genetic Algorithms with a Gene Bank. In: Miettinen, Kaisa et al. (eds.). Proceedings of EUROGEN99 – Short Course on Evolutionary Algorithms in Engineering and Computer Science. University of Jyväskylä, Department of Mathematical Information Technology, No. A 2/1999, pp. 162-170.

Tyni, T., Ylinen, J., 2001. Genetic Algorithms in Elevator Car Routing Problem. In: Spector, Lee et al. (eds.). Proceedings of the Genetic and Evolutionary Computation Conference (GECCO-2001). San Francisco, California, United States. July 7-11, 2001. Morgan Kaufman Publishers. pp. 1413-1422.

Ylinen, J., Tyni, T., 2001. Genetic procedure for multi-deck elevator call allocation, KONE Corporation, U.S. Patent No. 6 293 368. September 25, 2001.