Evolutionary bi-objective optimisation in the elevator
car routing problem

Tapio Tyni *, Jari Ylinen
KONE Corporation, R&D, P. O. Box 677, Hyvinkaa 05801, Finland
Accepted 13 August 2004
Available online 30 November 2004

Abstract

The paper introduces a genetic algorithms based elevator group control system utilising new approaches to multi-objective optimisation in a dynamically changing process control environment. The problem of controlling a group of elevators as well as the basic principles of the existing single-objective genetic elevator group control method are described. The foundations of the developed multi-objective approach, Evolutionary Standardised-Objective Weighted Aggregation Method, with a PI-controller operating as an interactive Decision Maker, are introduced. Their operation as a part of bi-objective genetic elevator group control is presented together with the performance results obtained from simulations concerning a high-rise office building. The results show that with this approach it is possible to regulate the service level of an elevator system, in terms of average passenger waiting time, so as to bring it to a desired level and to produce that service with minimum energy consumption. This has not been seen before in the elevator industry.

© 2004 Elsevier B.V. All rights reserved.

Keywords: Combinatorial optimisation; Control; Energy; Genetic algorithms; Transportation; Multi-objective optimisation; Elevator

1. Introduction

An elevator system is a vertical transportation system responsible to transport passengers, living, working or visiting in the building, comfortable and efficiently to their destinations. A properly sized and well behaving elevator system improves the usability of the building, thus not only affecting the comfort and service level directly experienced by the passengers, but also adding value to the building owner in terms of satisfied customers and tenants.

An elevator group or bank of elevators consists of a system where several elevators reside physically close together and respond to a common set of landing or hall call buttons located in the vicinity of the elevators at each floor. A passenger
orders a lift by pressing either an up or down call button, depending on the location of his/her destination floor with respect to the passenger’s arrival floor. When an elevator car arrives, the passenger enters the cage and gives his/her destination floor using car call buttons located in the cage. The order in which the elevator cars respond to the landing calls given by passengers—i.e. the routes of the cars—play a vital role in the performance of the vertical transportation system. In the elevator community, this problem is referred to as “landing call allocation problem” or “elevator dispatching problem” (Fig. 1).

The landing call allocation problem may be considered as a version of the travelling salesman problem (TSP)—a combinatorial optimisation task without a known method to be solved in polynomial time. In the TSP the salesman has a task to make a round trip around the cities, by visiting each city only once so that the distance he travels is the shortest possible. In the elevator car routing context the TSP problem can be converted to multiple travelling salesmen problem (MTSP), where salesmen (elevator cars) visit the cities (car calls and landing calls) so that each city (call) is visited only once and the cost function \(\sum C(S_i, C_i \cup L_i)\) is minimised. The partial cost \(C\) is gathered along the route when a set of car and landing calls \(C_i \cup L_i\) is visited by the elevator \(i\), starting the roundtrip from the elevator’s initial state \(S_i\). The size of the problem space is \(N_s = E^n\), where \(E\) is the number of elevators and \(n\) is the number of active landing calls. The problem is too large to be solved systematically, except only in the smallest elevator groups, so other methods have to be applied.

Examples of the methods that have been applied are intelligent agents [5,15], fuzzy logic [10,21], neural networks [8,14,17] and evolutionary and genetic algorithms [7,16]. The work done in the field can be basically divided into three branches: (i) the presented method is used to assist the existing dispatching algorithm; (ii) the method is used to tune the parameters of the existing dispatching algorithm and (iii) the presented method makes the actual control decisions and actions in real time. In [20] we presented a landing call allocation method based on Genetic Algorithms (GA) falling into category (iii) in the classification above. As far as the authors are aware of, the presented method is the first landing call allocation algorithm that performs true route optimisation, finding the optimal routes for each elevator in real time. The traditional methods, often strongly bound to expert knowledge and a priori heuristics of the elevator application domain, are able to provide only partial solutions. Their output is usually one allocated landing call per elevator, reasoned as the “best” by the method in question. Here we would like to distinguish between the terms “landing call allocation” and “elevator car routing”, so that the former is reserved for the methods providing partial solutions and the latter for methods performing true route optimisation according to some objective function. The method presented in the paper is the core of the TMS9900GA™ elevator group control system, part of the KONE Alta™, the third generation high rise elevator system from KONE intended for skyscrapers [12].

Fig. 1. Super-tall buildings rely on sophisticated, fast and quiet vertical transportation systems (computer rendering—X-ray picture of a super-tall building).
The optimisation in [20] was based on a single-objective function. In this paper, we extend the algorithm to a multi-objective optimisation (MO) capability and utilise two conflicting objectives—landing call waiting time and energy consumption of elevators along their routes. Due to the asymmetric properties of the elevator hoisting function, it is possible, by selecting the elevator car routes carefully, to conserve energy and still serve passengers in adequate time.

Special attention is paid to the issues of multi-objective optimisation in a real-time control application. The time frame available to obtain the control decisions is limited—routing decisions have to be made twice in a second. The constrained time frame imposes requirements regarding the computational efficiency of the optimisation algorithm. Most of the work in the evolutionary multi-objective (EMO) field deals with design type of applications where the time to obtain the solution is important but usually not a crucial requirement. The research has concentrated on developing methods that could obtain as many evenly distributed solutions as possible from the Pareto-optimal set of the problem, even from concave regions of the Pareto front. These algorithms are computationally expensive and the requirements concerning the control system hardware would be too demanding for commercial purposes.

Due to the processing time restrictions we present an approach based on the straightforward and computationally efficient Weighted Aggregation (WA) method. WA optimisation reduces the original MO problem to a single-objective optimisation problem, which have run times in $O(GN)$, where $G$ is the number of generations and $N$ is the population size, whereas the run times of the non-dominated sorting methods, like NSGA-II, are in $O(MGN^2)$, where $M$ is the number of objective functions [9]. The WA method presented in this paper is able to perform the bi-objective optimisation task with contradicting objectives in real time within the given 500-ms time frame. The drawback of the WA method is that it has problems in two areas: (i) the definition of the weight values depends on the optimisation problem and (ii) it cannot obtain solutions lying in the non-convex regions in the Pareto-optimal front [6].

To address problem (i), we present a general method which normalises the objective functions at the beginning of each evolutionary search, letting the Decision Maker (DM) simply express his preferences with weight values in the range $[0 \cdots 1]$ at any time. To deal with problem (ii), a control loop is established to ensure that the time averaged solutions meet the requirements given by the system operator. In the control loop a PI-controller acts as a DM adjusting the weights of the WA optimiser according to the deviations from the target passenger waiting time defined by the system operator. In the elevator world, the system operator is usually the building manager.

The ultimate goal behind the approach we present is to regulate the service level of the elevator system in terms of passenger waiting times or call time. During the design phase of the building, the elevators are sized to give appropriate service during intensive traffic peaks [2,11,19]. During other traffic periods, when traditionally only the passenger waiting time is optimised, the system provides inadequate good service at the expense of energy consumption and system wear. The bearing idea is to specify the average waiting time the system should maintain in all traffic situations—the elevator car routes should satisfy the specified average passenger waiting time with the least consumption of energy.

2. Principle of the genetic elevator car routing method

The principle of the Genetic Elevator Car routing problem is described to the extent that serves the purposes of the present paper dealing with multi-objective optimisation in the elevator car routing problem. For readers with further interest, a detailed description of the method and its performance can be found in [20].

Fig. 2 visualises how the elevator car routing problem is presented to the Genetic Algorithm. In the example the elevator group consists of two elevators. Elevator A is standing with doors open at floor 2. It has three car calls to floors 4, 5 and 7. Elevator B has no car calls and is standing with doors closed without a specific direction at floor
11. At this particular time the routing problem consists of four active landing calls that should be served by the elevators: two up calls at floors 4 and 8 and two down calls at floors 9 and 6. The elevator's system state consists of the general state of service (in service/out of service), car position, collecting direction (up, down), motional state (standing, moving, decelerating), car load, door state (opening, open, closing, closed) and the car call vector.

The chromosome is built up by taking the landing calls one by one and inserting them into the chromosome as “call genes”. The locus of a call gene represents the landing call and the allele of the gene represents the elevator car that should serve the call. This coding approach offers a natural way to meet the requirement that each call should be responded to only once by one of the elevators.

If an elevator has a direction, there is no problem—the direction can be fed straight away to the model of the elevator, as is the case with elevator A in this instance. When an elevator is standing at a floor without a specific direction (for example after it has served the last car call), it has an option to start collecting the landing calls in either up or down directions. The question is, which direction would be better? This problem can be resolved in an elegant way: if an elevator has no specific direction, a “direction gene” is included into the chromosome, specifying the direction in which the elevator should start. In fact, the direction gene specifies indirectly the first landing call in the sequence of calls to be served that a single elevator should obey.

The optimisation is performed continuously in real time twice a second or immediately if the landing call situation changes. As the traffic situations vary greatly in the course of the day, the elevator car routing problem is an optimisation problem of dynamic nature. During the same optimisation problem, the system state changes as the elevators proceed along their routes. However, despite the changes in the system state, the genetic search should provide the same route solution it did the first time the new problem occurred, i.e. new active landing calls appeared. This also applies to the type of new optimisation problems where an elevator has served a call and the call disappears—the rest of the routes should remain as they were in the original problem. Genetic search has random properties, and in the case of multi-modal problems, which is what this problem appears to be, there is no guarantee that the search ends up with the same solution each time the optimisation is run. Here we have improved the stability of the

---

Fig. 2. Illustration of the genetic elevator car routing method.
solutions by initialising one of the chromosomes of the initial population with the route solution from the previous optimisation cycle. That particular chromosome is superior compared to the other, randomly generated chromosomes of the initial population, biasing the search towards the previous solution. The stability issues are also studied in detail in [20].

The route alternatives for the cars represented by the chromosome are fed to a cost evaluation procedure, which decodes the chromosome and assigns the landing calls and the possible direction gene to the appropriate elevator models. Each elevator model then forms its route in the building on basis of the system state, the landing calls assigned to it and the behavioural rules an elevator has to obey. The essential rule is the collective service principle—the elevator can reverse its direction only when it has no car calls in its direction of movement.

The cost of the route alternative is returned to the optimiser. The traditional objective functions have been the sum of call times or estimates of the passenger waiting times. Once the route has been formed in the elevator model it could also be used to evaluate some other objectives—like energy consumption of the route alternative, among others.

3. Energy consumption of the elevator hoisting function

The components of the hoisting mechanism of an elevator are so designed that the system is in balance when the car is half-loaded and at the same height $h$ as the counterweight

$$m_{cw} + \frac{1}{2}m_{ropes}(h) = m_{car} + \frac{1}{2}m_{L \text{max}} + \frac{1}{2}m_{ropes}(h),$$

(1)

where $m_{L \text{max}}$ is the maximum rated load; refer to Fig. 3. When the car and counterweight are located at different heights in the shaft, the ropes become unbalanced. In mid and high-rise elevators with shaft lengths up to several hundreds of meters, the imbalance due to ropes has to be compensated with another set of ropes connecting the car and counterweight through the pit. In that case, Eq. (1) reduces to

$$m_{cw} = m_{car} + \frac{1}{2}m_{L \text{max}}.$$  

(2)

In low-rise elevators, the rope masses are much smaller compared to the other masses in the hoisting system and Eq. (2) can be applied with sufficient accuracy as well.

When an elevator car runs from a floor at height $h_1$ to some other floor at height $h_2$ the potential energy in the system changes:

$$\Delta E = mg(h_2 - h_1) = mg\Delta h.$$  

(3)

In Eq. (3), the mass $m$ is the static mass balance of the system:

$$m = m_{car} + m_L - m_{cw} = m_L - \frac{1}{2}m_{L \text{max}}.$$  

(4)

Consider an empty car moving downwards. The counterweight is heavier than the car and the hoisting motor has to apply energy to the system in order to increase the system’s potential energy. In the opposite case, when an empty car moves upwards, the heavier counterweight pulls the car and the motor has to brake, i.e. it functions as a generator. Depending on the motor drive electronics, the released potential energy is either wasted into a braking resistor or, as in the more advanced systems, it can be returned to the power supply network. When the car is fully loaded, the directions are reversed—moving the car upward consumes
energy and moving it downward restores the potential energy. In fact, passengers transported into the building conserve energy, which in turn is released when passengers leave the building. All the potential energy bound to the passengers cannot be restored because of the mechanical and electrical losses in the transportation system.

Fig. 4 shows an example representing an elevator travelling upwards (2–15 seconds) and then downwards (30–43 seconds) with a fully loaded car. The nominal speed of the hoisting system is 3.5 m/s, acceleration 1.2 m/s² and maximum load 1500 kg or 21 persons. This 13-floor hoisting system is used in the examples throughout this paper.

If the distance is long enough so that the constant speed phase can be attained, a trip between floors includes three phases: acceleration, constant speed and deceleration. In Fig. 4, moving the fully loaded system upwards with a constant speed of 3.5 m/s requires 30 kW of power from the hoisting motor during the period from 5 to 9 seconds. Whereas, under the same conditions, during the

![Fig. 4. Car acceleration, speed, position and energy consumption for an upward trip and then a return trip down to the starting floor with a full load of 21 persons.](image)

![Fig. 5. Energy consumption of an elevator during upward (left) and downward (right) movement as a function of running distance. Car load 100% equals 21 passengers.](image)
downward constant-speed portion of the trip, the motor power is \(-18\) kW, i.e. it functions as a generator supplying energy to the power line or to a braking resistor, depending on the technology used in the motor drive system. In the end, the net energy consumed during this round trip is 55 Wh.

Fig. 5 shows the energy surfaces of the example system as a function of car load (in persons) and running distance in floors.

Considering the energy surfaces in Fig. 5 and the simple routing problem example in Fig. 6, it is quite clear that there may be noticeable differences in the consumption of the energy depending on the routes of the elevator cars serving the passengers. When the normal passenger waiting times are to be optimised, the optimal routing is as in Fig. 6 left, and in respect of energy consumption the optimum routes are as in Fig. 6 right. Generally, when optimising the energy consumption of the elevator car routes, the cars should travel upwards as empty as possible and downwards as full as possible.

4. Evolutionary standardised—objective weighted aggregation method (ESOWA)

The literature on evolutionary multi-objective optimisation deals largely with problems of analysis and design type, e.g. optimising some mechanical structure. The focus can be seen e.g. in [4]. These problems are off-line in nature, the computational expense of the algorithm is important but not the most crucial aspect.

Instead, in real time control type optimisation problems, the time frame available to achieve the solution is limited and crucial. For example, in the case of our elevator group control application, routing decisions have to be made twice a second. With the popular Pareto-dominated EMOs, the processing times comply with the formula \(O(MGN^2)\), where \(M\) is the number of objective functions, \(G\) is the number of generations and \(N\) is the population size [9]. As the Weighted Aggregation (WA) method is essentially a single-objective optimisation method, the processing times follow the formula \(O(GN)\). Using Pareto-dominated sorting algorithms would yield \(O(MN)\) longer processing times as compared to the WA algorithms, which would be unacceptable in this real time application.

The WA method is classified either a posteriori or a priori method, depending on how it is applied. In this paper, we consider it as an a priori method as the Decision Maker (DM) balances the importance of each objective function in terms of weight coefficients and the method then returns a solution from that region in the Pareto front. In the WA method, a problem to be solved is

\[
\text{minimize } \left\{ \sum_{i=1}^{n} w_i f_i(x) \right\}
\]

subject to \( x \in D \),

where \( w_i \geq 0 \forall i = 1, \ldots, k \) and \( \sum_{i=1}^{k} w_i = 1 \) and \( D \) is the feasible region of the solutions [6]. The method returns the multi-objective problem to a scalar optimisation one, and all the developed evolutionary methods in that field are readily available.

To obtain the non-dominated set of decision vectors \( x \), the optimisation problem is normally run with different linear combinations of weights \( w_i \). A problem arises when the ranges of each objective function differ significantly or are not known beforehand or the optimisation task changes constantly. To balance the effect of each objective function on the aggregated cost \( C \), a priori knowledge about the problem is needed to adjust the weights to proper ranges. In the off-line type applications this is tolerable, as it is possible to experimentally learn the ranges of each objective function and adjust weights accordingly. In
real-time control applications in dynamical environments, there is no chance for such experiments as the system is running on its own, making control decisions without human intervention.

A technique to overcome the difficulties mentioned above is to consider the objective function $f_i$ as a random variable and apply standardisation. If an objective function $f_i$ has a distribution $\Delta_i(\mu_i, \sigma_i^2)$, then a standardised objective function

$$\phi_i = \frac{f_i - \mu_i}{\sigma_i}$$

has a distribution $\Delta(0,1)$. The underlying mean $\mu$ and variance $\sigma^2$ of the objective functions are not (normally) known a priori. Instead, they have to be estimated using their counterparts sample mean $m_i$ and sample variance $s_i^2$,

$$m_i = \frac{1}{N} \sum_{j=1}^{N} f_{ij},$$

$$s_i^2 = \frac{1}{N-1} \sum_{j=1}^{N} (f_{ij} - m_i)^2,$$

where $N$ is the sample size. The standardisation is now formulated as

$$\phi_i = \frac{f_i - m_i}{s_i}$$

and the optimisation problem is now

minimize $\left\{ \sum_{i=1}^{n} w_i \phi_i(x) \right\}$

subject to $x \in D,$

where again $w_i \geq 0 \ \forall i = 1, \ldots, k \ \& \ \sum_{i=1}^{k} w_i = 1.$

With the evolutionary algorithms, the sampling is a built-in feature of the algorithm. The initial, (uniformly) randomly generated first generation of decision vectors $x$ take $N$ random samples from the objective function space for each $f_i : Z^n \rightarrow R$. The sampling and normalisation approach with the evolutionary algorithms with population size $N$ is straightforward as illustrated in Table 1.

As a random variable, the sample mean $m$ also has a mean $\mu_m$ and a standard deviation $\sigma_m$. Function $f : Z^n \rightarrow R$ is considered as a random variable with mean $\mu$ and standard deviation $\sigma$. The

<table>
<thead>
<tr>
<th>Table 1</th>
</tr>
</thead>
<tbody>
<tr>
<td>Steps of the evolutionary standardized objective weighted aggregation method</td>
</tr>
<tr>
<td>1. Create the initial population of $N$ decision vectors randomly, compute the sample mean $m_i$ and variance $s_i^2$ in Eq. (7)</td>
</tr>
<tr>
<td>2. Compute the standardized objective functions $\phi_i$ in Eq. (8)</td>
</tr>
<tr>
<td>3. Compute the aggregated sum in Eq. (9)</td>
</tr>
<tr>
<td>4. Apply the genetic operators (selection, crossover and mutation) specific to your (single-objective) evolutionary optimiser to create the next generation of population</td>
</tr>
<tr>
<td>5. If not converged, go to the step 2</td>
</tr>
</tbody>
</table>

Central Limit Theorem [13] states that if the sample size $N$ is large (say $N \geq 30$) and the population from which the samples are drawn is large in comparison to $N$, then the sample mean $m$ is approximately normally distributed with mean $\mu_m = \mu$ and $\sigma_m = \sigma / \sqrt{N}$. The analysis of variance is more complicated but has similar behaviour. If needed, the confidence interval of the sample mean and variance can be improved by increasing the population size $N$ for the initial generation.

Fig. 7 illustrates the search behaviour with a simple and smooth convex test function in Eq. (10). The objective functions operate in different ranges, in spite of the fact that the search is able to find solutions from the convex Pareto front at roughly equal distances when the weights move with linear combinations of $w_1 = [0, 0.1, \ldots, 1]$ and $w_2 = [w_1 - 1]$. Note that the entire objective function normalisation is hidden from the user, taking place “on the fly” inside the algorithm during the search. The basic genetic algorithm was a simple GA with integer genes coded to range $-1.5 / \sqrt{10} \leq x_i \leq 1.5 / \sqrt{10}$, population size 100, uniform crossing over and elitist selection with 20 parents.

\[
\begin{align*}
    f_1(x_1, x_2, \ldots, x_{10}) &= 0.01^* \left(1 - \exp \left(-\sum_{i=1}^{10} (x_i - 1 / \sqrt{10})^2\right)\right)^3, \\
    f_2(x_1, x_2, \ldots, x_{10}) &= 100^* \left(1 - \exp \left(-\sum_{i=1}^{10} (x_i + 1 / \sqrt{10})^2\right)\right)^5.
\end{align*}
\]
5. Passenger traffic in the building

Typically, in residential and office buildings, the daily passenger traffic repeats itself regularly from day to day. The traffic can be divided into three main components: incoming traffic from the entrance floor(s) to different floors in the building, outgoing traffic flows from the ordinary floors to the entrance floor(s) and interfloor traffic between ordinary floors (entrance floor(s) excluded). Fig. 8 shows a measured example of a traffic pattern from an office building. Mathematically, the traffic is modelled as an unstationary Poisson process [3]

\[ P\{k \text{ arrivals in } \Delta t\}(t) = \frac{(\lambda(t)\Delta t)^k}{k!} e^{-\lambda(t)\Delta t}, \]  

(11)

where \( \lambda(t) \) is the time-varying mean of a traffic component. Eq. (11) gives the probability for the event that \( k \) passengers will arrive during a (short) time period \( t + \Delta t \), over which the \( \lambda(t) \) can be considered as constant.

Traditionally, the operation of the landing call allocation algorithms is tuned to fit the prevailing traffic type detected by the elevator group control system. The basic traffic types that are normally detected are normal traffic, incoming peak, outgoing peak and two-way peak. The landing call allocation algorithm then uses pre-set parameter sets for the particular, detected traffic type, adapting its operation and control strategies to the current traffic situation. Elaborated methods have been developed to get even more detailed traffic types, e.g. in [18], where reasoning with fuzzy sets is utilised to detect 36 different traffic types. In addition, traffic intensity information from each floor can be gathered to adjust the operation of the control system.

The problem with this approach is how to define the actual parameter values for each traffic type for the building and the elevator system in
it. One approach is offline simulation. The building and the elevator group are configured into the simulator [18,19]. If the building is in the design phase, then the traffic can be estimated from the building’s design parameters. The simulation is carried out for each detected traffic type and the parameters are tuned manually to give the best performance. That is a drawn-out and tedious process [1].

When the optimisation process is automated, it also enables online tuning of the parameters, reducing the sensitivity to changes in the environment and system. However, the drawback is that the on-line simulation is a computationally intensive task consuming resources from the control system’s Decision-Making tasks. Even if the resources were available, the principal question still remains—what is the basis to choose the control parameters for the traffic situation at hand? In prior art, the definition has been more qualitative than quantitative in nature. As an example, relative values among the parameter set may be defined so that during light and medium traffic the emphasis is on energy consumption while during heavy traffic peaks the focus is on passenger service. Nevertheless, no method of how these more qualitative settings are tied to actual quantitative numbers describing the service level of the transportation system has been seen so far.

6. The control method

A severe limitation of the WA method is its disability to deal with the non-convex regions of the Pareto front [6]. The non-convexity is also present in the elevator car routing problem. Fig. 9 shows an elevator group of 7 elevators, an optimisation problem instance and the corresponding fitness space. The example implies that the Pareto front in this application may contain none, one, two, or even more local concave regions not reachable by the WA method.

In order to provide good service, the elevating guidelines recommend that an elevator system should be able to transport 12–15% of the building’s population within a 5-minute period. That is, the passengers can be transported into the building in 42–33 minutes. The traffic applied in these calculations is the pure incoming traffic peak as the operation of the system is not influenced by the properties of the car dispatching function, thus allowing mathematical analysis of the elevator system’s traffic handling performance. As the elevator system is sized to give good service during the intensive morning incoming traffic peaks, it can be stated that the service during normal and light traffic is “unnecessarily good”. The system has excessive resources for the traffic outside the peak periods and produces inappropriately short waiting times at the expense of, for example, energy consumption and system wear.

The control principle is to find routes for the elevator cars that will satisfy the given target passenger waiting time with minimum consumption of energy, i.e.

\[
\text{minimize } \{f_1(x), f_2(x)\}
\]

subject to \(x \in D\),

\[
x^* = \{x \in P^* | (f_1(x) - f_1^*)^2 = \min\}. \tag{13}
\]

In the Decision Maker’s utility function Eq. (13), \(f_1^*\) is the specified target for the average call time the system should maintain in the prevailing traffic conditions and \(x^*\) is the decision vector or allocation decisions from the set of non-dominated solutions \(P^*\) forming the Pareto front. In order to obtain the \(P^*\) with the WA method, the optimisation should be run with a number of linear combinations of weights \(w_\alpha\) whereupon Eq. (13) should be applied. To reduce the computational burden and further, to compensate the difficulties of the WA method with the concave Pareto-regions, we take an approach where the optimisation is executed only once per control cycle. To reach the correct regions of the Pareto front, the weights \(w_i\) are adjusted continuously during the course of the operation so that the time average of term \((f_1(x) - f_1^*)^2\) in Eq. (13) is minimised. During each control cycle, a dedicated controller acts as a Decision Maker comparing the differences between the predicted average call time \(f_1(x^*)\), produced by the elevator system model in an evolutionary optimiser, to the target value \(f_1^*\). The DM thus guides the optimiser by adjusting the objective function weights according to the PI-control rule in order to satisfy Eq. (13) over
**Fig. 9.** A traffic situation (left) and corresponding fitness spaces (right). Random initial values (black crosses), 64,000 random values (grey diamonds), best fitness from each generation (red circles) and final solutions found by ESOWA (black dots). There are two locally convex regions on the Pareto front, pointed by arrows. (For interpretation of the references in colour in this figure legend, the reader is referred to web version of this article.)

**Fig. 10.** Overall system structure. A PI-controller (Proportional and time-Integral terms) acts as a Decision Maker guiding the Optimiser to provide a specified service level in terms of average call time. The predicted Average Call Time $f_1(x^*)$ is obtained from the Optimiser's system model as a “side product” of the optimisation.
course of time. Block diagram of the approach is shown in Fig. 10.

The \( k \)th predicted average call time \( f_i(x^i_k) \) from the Pareto front is smoothed in the estimator block with

\[
\hat{f}_{1,k} = \hat{f}_{1,k-1} + (\hat{f}_{1,k-1} - f_i(x^i_k)) \cdot G_E
\]

and compared in the error block to the target service level \( f^*_i \) defined by the system operator, usually the Building Manager taking care of the building’s facilities. The difference \( e_k = f^*_i - \hat{f}_{1,k} \) between the target and estimated average call times is input to the controller block. The control rule is the standard PI. The output \( u_k \) of the controller is the sum of term \( u_P \) proportional to the error \( e(t) \) and \( u_I \) the time integral of error \( e(t) \), in the continuous time domain

\[
u(t) = u_P(t) + u_I(t) = G_P e(t) + G_I \int_0^t e(t) \, dt.\] (15)

It is characteristic of the closed loop controlled system that the controller tends to keep the difference between the target and the actual value close to zero. This action takes place regardless of whether the deviation from the desired value has occurred due to reasons internal or external to the controlled process.

A natural choice for the integrator gain \( G_I \) comes from the nature of the passenger traffic. As the traffic intensity unit in the elevator world is “passengers in five minutes”, \( G_I \) may be selected so that the integrator provides a time constant of the same order of 5 minutes. The selection of the proportional gain \( G_P \) is more problematic. The total gain of the closed loop system is the controller gain \( G_P \) multiplied by the variable system gain \( G_S = \frac{d f_i}{d w_1} \). \( G_S \) depends on the region of the Pareto front the system is operating in. If the total gain in the control loop is too large, the series

\[
\lim_{k \to \infty} (f_i(x^i_k) - f_i(x^i_{k-1})
\]

does not converge and the system is unstable. To avoid instability, conservative values for the proportional gain \( G_P \) have to be used; the integral term will have the main responsibility in the control loop. The estimator gain \( G_E \) is selected so that it smoothes the predicted values without introducing too much delay.

7. Results

As it is not possible to obtain comprehensive data in a controlled manner from elevators running in a real building, a building simulator with simulated passenger traffic and elevators is a standard development tool in elevator industry [19,18]. The example building and elevator system we use here is the elevator group of 7 elevators and 19 floors illustrated in Fig. 9. The specifications of the hoisting system were given at the beginning of the paper. The nominal transportation capacity of the system is 200 passengers in 5 minutes, which equals 13.1% of the building population and is classified as good [11]. Three independent simulation series were run with pure outgoing traffic. The results shown here are averages of the three series. Pure outgoing traffic was selected because in that traffic type the optimiser has most freedom to arrange the routes, as there is no constraining car calls, revealing most clearly the capabilities of the control principle. Within one simulation series, the traffic intensity was increased from 2% of the building population to 20%, in 2% steps. Each intensity step was run for 30 minutes.

Fig. 11 shows as an example the behaviour of the DM-control loop during one simulation run with 8% intensity. The horizontal axis is time up to 1800 seconds. The topmost curve shows the number of active landing calls. Presented in the low section of the graph are call time estimates from the estimator, realised call times, and 5 minutes moving average of the realised calls. In the middle is the weight \( w_1 \) i.e. the DM’s preference for call time importance for the optimiser.

A representative example of the behaviour is between 800 and 900 seconds. A burst of 11 active landing calls appears in the system and the call time estimator indicates that the average call time will increase. Because of that, the Decision Maker increases the importance of call times in the optimisation. Once that burst has been handled, the call time predictions from the estimator tend to decline, but DM responds by increasing the preference of energy consumption. During the operation, the 5 minutes moving average of the realised call times follows very closely the 20-seconds goal for the service level. This implies that
the system model in the optimiser the routing actions are based on describes the system with high fidelity.

That the system is under good control can be seen from the realised call time distribution in the time domain behaviour in Fig. 11 and in the cumulative distribution histogram in Fig. 12. About 95% of the realised call times fall below 50 seconds, and no excessively long times appear. The average call time over the whole simulation run is 20.7 seconds with a narrow deviation as seen from the distribution of the 300 seconds moving average histogram in Fig. 12 left.

The behaviour of the call time weight $w_1$ as a function of traffic intensity is shown in Fig. 12 right. The graph is the average over the three 30 minutes test runs. The system reaches its capacity limits with the given service level of 20 seconds at 18% traffic intensity, after that the system only optimises the waiting time.

---

**Fig. 11.** Behaviour of the Decision Maker's control loop at a traffic intensity of 8%.

**Fig. 12.** Cumulative probability distribution of realised call times (left). Average value of weight $w_1$ as a function of traffic intensity (right). Averaged over three 30-minute simulation runs.
Fig. 13 collects the main results from the three simulation series with three different control strategies. The horizontal axis is traffic intensity as a percentage of the building population. The three different routing strategies are pure call time optimisation with $f_1^* = 0$ seconds, pure energy optimisation with $f_1^* = \infty$ seconds, and adequate service level with $f_1^* = 20$ seconds.

Considering the call times, the figures produced by pure energy optimisation would be definitely useless in practice. The standard approach, pure call time optimisation, pushes the figures down to 10 seconds during low traffic intensities. These two methods together set the limits for the possible system operating range. It is remarkable how well the third control strategy, in which we are specially interested here, is able to maintain the given goal of 20 seconds.

The middle section of Fig. 13 is of special interest. It shows what happens to the energy consumption when the service level requirement is relaxed. The power figures are obtained by dividing the cumulated energy consumption during the test run by the test run length, i.e. $P = \text{Energy}/\text{time}$. The power figures thus represent the average power level the elevator group hoisting function has used during the test run.

The reduction of energy consumption and the required power level are dramatic in the low and medium traffic intensities. The best case is at an intensity of 6%, where the hoisting power drops 14kW—from 19 to 5kW. The extra seconds provided to the passengers by the standard call time optimisation are really expensive, as shown in the lowest section of the graph. The small difference in the 20% intensity is explained by the limit of 0.95 for the weight $w_1$ (see Fig. 10). The system is never allowed to go to the pure call time optimisation mode, not even at the highest traffic intensities, but the energy aspect is always kept more or less in consideration.

Fig. 13. Performance results of three different control strategies: pure energy optimisation (circle), pure call time optimisation (box) and specified service level 20 seconds (diamond).
Fig. 14 shows the positive side effects of the approach. There is as dramatic a fall in the running distance of the elevators as there is in the hoisting function’s power level. Also the number of starts is reduced significantly.

As both the running distance and the number of starts have been decreased, it means the optimiser uses the resources in a way that allows more passengers to be collected into the car during one round trip. This is in line with the properties of

![Graph](image-url)

**Fig. 14.** Total running distance and number of starts of the elevator group. Average of three 30 minutes test runs.

![Graph](image-url)

**Fig. 15.** Average call time over a working day in an office building with traffic as shown in Fig. 8. Pure waiting time optimisation (down), pure energy optimisation (up) and waiting time optimisation with goal of 20 seconds (middle).
the outgoing traffic: the cars should be filled as full as possible in order to release the potential energy that each passenger possesses.

The running distance and number of starts also have a system wear and maintenance aspect. The running distance affects directly, for example, the wear of the rope assembly. After each start to a destination floor, the elevator stops and the doors will operate. The doors are the most sensitive component in the elevator system. Each removed start is also a removed door operation with a positive impact on the door maintenance needs.

Results from daily traffic are shown in Fig. 15. Outside the morning, lunch and outgoing peaks, the average call times in the ordinary waiting time optimisation are around 10 seconds. In these periods of time it is possible to reduce the consumption of energy. Running the system with the 20-second target average call time reduces the daily energy consumption from 288 kWh consumed by pure waiting time optimisation down to 231 kWh, which corresponds to a $20\%$ relative decrease. At the same time, the running kilometres are reduced by $22\%$ and the number of starts by $9\%$.

8. Discussion

The approach to controlling the elevator group so that the passenger service level is regulated is a new concept in the elevator industry. In the elevator industry, controlled systems have been used for a long time. The hoisting motor does not run with full power when the elevator is driven from one floor to another. Instead, there is a control system that accelerates and decelerates the car gently to assure a good ride comfort. The door motor does not open and close doors as quickly as possible. Instead, there is a control system which drives the door according to a smooth speed curve. Similarly, the group control system should not operate with full power all the time. Instead, the group control system shall provide adequate passenger service with minimum energy consumption. With the presented approach, the elevator group performance is under full control and can be defined explicitly and understandably simply with one parameter, the average call time or passenger waiting time. As a bonus, the weight $w_1$ varying in range from 0 to 1, can be presented to the system operator as a relative load indicator.

The presented ESOWA enhancement to the WA method—the standardisation of the objective functions based on the statistical properties of the initial population—is of a general nature and can be used in any application if the problem nature allows the straightforward WA method to be applied at all. With ESOWA, the DM does not have to worry about the objective function ranges, but all the objective function weights may be given in the range of 0–1. In this application, where every building has its own characteristics and every optimisation task differs from its predecessor, this is an essential benefit and requirement.

The orthodox approach to tackle the MO problem is to utilise some of the developed non-dominance rank based evolutionary MO method, obtain a large set of non-dominated solutions and pick the one which best satisfies the DM utility function. The big obstacle with this approach is the computational load compared to the available time frame to obtain the control actions and computing resources in the target control hardware. In addition, these methods are not totally free of application-dependent tuning parameters. Moreover, in the end, only one solution is used and the others are discarded. In a way, this is a waste of scarce computational resources in the target hardware. Because of that, the decision to study the use of the computationally effective WA method was taken.

Now, the question still remains why the WA method gives good results here although, referring to Fig. 9, the problem clearly has concave regions and the WA method is able to find only a few distinct solutions and apparently shouldn’t work properly. If the optimisation task consisted of this single one, then the performance would be intolerable. However, here the optimisation task is an endless procedure triggered twice a second to solve continually varying problems. There is no final answer and no final Pareto-set that should be obtained. In these circumstances, in the long run, although there are regions that are hidden from the straightforward WA method, the DM is able to guide the optimiser and the process so that the set goal will be satisfied sufficiently. The
standard interactive process described in MO literature to solve one optimisation problem is here spread over the time. In that context, the success of applying the WA method to solve a problem with apparently concave regions can be understood.

9. Conclusions

The present paper introduced an approach to the control type to be used in optimisation applications. The focus in this paper was to enhance the former single-objective elevator group control method to bi-objective control with two conflicting objectives—the consumption of energy of the elevator routes and the system service level in terms of average call time. The bi-objective optimisation is performed in real time by the Evolutionary Standardised Objective Weighted Aggregation method presented here. Because of the real time and stand-alone requirements, special attention is paid to the computational efficiency and automatic scalability to the problem being optimised. An inner control loop was established to regulate the service level of the elevator group. Within the inner loop, a PI-controller interacts as a Decision Maker with the optimiser. Simulations with high-rise office building as a test case ensured the feasibility of the approach. The results indicate that the approach is able to maintain the specified passenger service level with minimum consumption of energy. The reduction in the energy consumption comes from asymmetric properties of the elevator's hoisting system, potential energy the passengers posses and careful and economising usage of transportation resources. The control approach also provides positive side effects on system wear and maintenance costs. Now, with the presented approach, for the first time in the elevator industry, it is possible to specify and maintain the service level the transportation system should provide in an understandable and simple way.

References


