# Algorithm Based Validation of a Simplified Elevator Group Controller Model

Thomas Beielstein* Sandor Markon† Mike Preuss*

*Universtität Dortmund
D-44221 Dortmund, Germany
{tom,preuss}@LS11.cs.uni-dortmund.de

†FUJITEC Co.Ltd. World Headquarters
28-10, Shoh 1-chome, Osaka, Japan
markon@rd.fujitec.co.jp

## 1 Introduction

Elevators play an important role in today's urban life. The central part of an elevator system, the elevator group controller, assigns elevator cars to service calls in real-time while optimizing the overall service quality, the traffic throughput, and/or the energy consumption. The elevator supervisory group control (ESGC) problem can be classified as a combinatorial optimization problem [Bar86, SC99, MN02]. Its behavior reveals the same complex behavior as many other stochastic traffic control problems, i.e. materials handling systems (MHS) with automated guided vehicles (AGVs).

Due to many difficulties in analysis, design, simulation, and control, the ESGC problem has been studied for a long time. First approaches were mainly based on analytical approaches derived from queuing theory, whereas nowadays computational intelligence (CI) methods and other heuristics are accepted as state of the art [CB98, MN02, SWW02].

In this article we will propose a validation methodology for a simplified ESGC system, the sequential ring (S-ring). The S-ring is based on a neural network (NN) to control the elevators. Some of the NN connection weights can be modified, so that different weight settings and their influence on the ESGC performance can be tested. The performance of one specific weight setting $\vec{x}$ is based on simulations of specific traffic situations, which automatically lead to stochastically disturbed (noisy) fitness function values $\tilde{f}(\vec{x})$. The determination of an optimal weight setting $\vec{x}^*$ is difficult, since it is difficult to find an efficient strategy that modifies the weights without generating too many infeasible solutions, and to judge the performance or fitness $f(\vec{x})$ of one ESGC configuration.

The S-ring was introduced as a benchmark problem to enable a comparison of ESGC algorithms, independently of specific elevator configurations [BFM00, Bey01]. Results from the S-ring, obtained with low computational costs, should be transferable to more complicate ESGC models.

In the following, we will present different techniques to answer the question whether or not the S-ring is a simplified, but valid ESGC simulation model. We propose a new validation methodology that takes the optimization algorithm for the simulation model into account. After a few runs of the optimization algorithm on the simplified simulation model, we can determine good parameter settings for the optimization algorithm, that are also applicable to the complex simulation model. Thus, improved algorithm parameter settings obtained from simulation results on the S-ring should be transferable to real ESGC problems. S-ring simulations might give valuable hints for the optimization of highly complex elevator group controller optimization tasks.

The rest of this paper is organized as follows: in Section 2, the ESGC problem is introduced. Section 3 discusses different validation techniques. Section 4 gives a summary and an outlook.

# 2 The Elevator Supervisory Group Control Problem

## 2.1 Elevator Control

We introduce one specific instance of the ESGC problem, a so-called destination call system: in contrast to traditional elevators, where customers only press a button to request up or down service and choose the exact destination from inside the elevator car, a destination call system enables the customer to choose the desired destination at a terminal before entering the elevator car [BEM03]. The following investigations are based on a neural network based controller, developed by Fujitec, one of the world's leading elevator manufacturers. This controller is trained by use of a set of fuzzy controllers, each representing control strategies for different traffic situations [Mar95].

The concrete control strategy of the neural network is determined by the network structure and neural weights. While the network structure as well as many of the weights are fixed, some of the weights on the output layer, which have a major impact on the controller's performance, are variable and therefore subject to optimization. Thus, we are looking for an algorithm to optimize the variable weights of the neural controller. The controller's performance can be computed by the help of a discrete-event based elevator group simulator developed and provided by Fujitec. This ESGC simulation model will be referred to as the 'lift model' (or simply 'lift') throughout the rest of this paper.

Unfortunately, the resulting response surface shows a couple of characteristics which makes the identification of globally optimal weights difficult if not impossible. The topology of the fitness function can be characterized as highly nonlinear and highly multi-modal. It is randomly disturbed due to the nondeterminism of service calls, and dynamically changing with respect to traffic loads. Thus local measures such as gradients can not be derived analytically.

Furthermore, the maximum number of fitness function evaluations is limited to the order of magnitude $10^4$, due to the computational effort for single simulator calls.

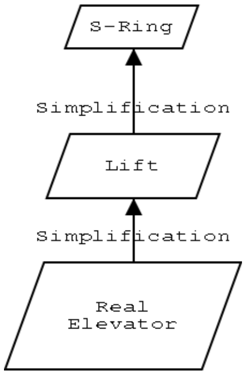


Figure 1: Relationship between the S-ring, the lift model and real elevator systems.

### 2.2 The Lift Model Objective Function

The objective function for this study is the average waiting time of all passengers served during a simulated elevator movement of two hours. Different traffic patterns occur during this time-period: up-peak (morning rush hour), two-way (less intense, balanced traffic during the day), and down-peak traffic (rush hour at closing time).

An evolution strategy (ES) was chosen to determine optimal NN weights [BS02, BEM03]. For the comparison of different ES parameter settings the best individuals produced by the ES were assigned handling capacities at 30, 35, and 40 seconds. A handling capacity of $n$ passengers per hour at 30s means that the elevator system is able to serve a maximum of $n$ passengers per hour without exceeding an average waiting time of 30s. These values were created by running the lift simulator with altering random seeds and increasing passenger loads using the network weights of the best individuals found in each optimization run. Finally, to obtain a minimization problem, the handling capacities were averaged and then subtracted from 3000 pass./h. The latter value was empirically chosen as an upper bound for the given scenario. The resulting fitness function is shown in Eq. 1 and is called 'inverse handling capacity' in the following.

$$F(x) = 3000.0 - \overline{f}_P(\vec{x}), \tag{1}$$

where $\overline{f}_P$ is the averaged handling capacity (pass./h), $P$ is the parameter design of the evolution strategy optimization algorithm (cf. Eq. 6), and $\vec{x}$ is a 36 dimensional vector that specifies the NN weights.

### 2.3 The S-ring Model as a Simplified ESGC Model

The S-ring can be seen as a simplified and easily reproducible ESGC model. It is based on the Kac ring, see e.g. [Gou95]. The S-ring can be solved exactly for small problem sizes, while still exhibiting non-trivial dynamics. The main differences between ESGC models and the S-ring can be summarized as follows: Elevator cars in the S-ring model have unlimited capacity, and passengers are taken, but not discharged. The running directions of the cars are only reversed at terminal floors. All floors are indistinguishable: there are identical passenger arrival rates on every floor, and identical floor distances. The cars use uniform running and stopping times, and the whole model uses discrete time steps. Furthermore, sequential state transitions are performed [MAB+01].

The S-ring can be seen as a one-dimensional cellular automaton[CD98]: The state at time $t$ is given as

$$[s_0(t), c_0(t), \ldots, s_{n-1}(t), c_{n-1}(t)] \equiv \mathbf{x}(t) \in \mathbf{X} = \{0,1\}^{2n}.$$

There are $n$ sites, each with a 2-bit state $(s_i, c_i)$, and with periodic boundary conditions at the ends. $s_i$ is set to 1 if a server is present on the $i$th floor, otherwise it is set to 0. The same applies to the $c_i$ bits: they are set to 1 if at least one customer is waiting on the $i$th floor. Instead of using synchronous updating at all sites independently, one updating cycle is decomposed into $n$ steps as follows: The state evaluation is sequential, scanning the sites from $n-1$ to 0, then again around from $n-1$. At each time step, one triplet $\xi \equiv (c_i, s_i, s_{i+1})$ is updated, the updating being governed by the stochastic state transition rules, and by the 'policy' $\pi : \mathbf{X} \to \{0,1\}$. A new customer arrives on the $i$th floor with probability $p$, and $m$ different elevator cars are considered [MAB+01].

### 2.4 The S-Ring Model Objective Function

The S-ring model can be used to define an optimal control problem, by equipping it with an objective function $Q$ (here E is the expectation operator):

$$Q(n, m, p, \pi) = \mathrm{E}\left(\sum c_i\right). \tag{2}$$

Thus $Q$ can be read as the expected number of floors with waiting customers. For given parameters $n$, $m$, and $p$, the system evolution depends only on the policy $\pi$, thus this can be written as $Q = Q(\pi)$. The optimal policy is defined as

$$\pi^* = \arg\min_{\pi} Q(\pi). \tag{3}$$

The basic optimal control problem is to find $\pi^*$ for given parameters $n$, $m$, and $p$. The performance of a particular policy cannot be determined exactly, it must be estimated. Problems related to optimization via simulation are discussed in [BINN01].

## 3 The S-Ring Model as a Valid ESGC Model

### 3.1 Context Description

In the following we will present standard validation techniques for simulation models. We will differentiate between model verification and model validation. Although verification and validation of simulation models are related in some sense, we will consider validation only [LK00, BINN01]. Classical validation processes are used to produce a model that represents a given system behavior. This model should be accurate enough that it can be used as a representative of the real system.

We will extend these techniques by introducing a new approach, that takes the choice of an optimization algorithm into account. Our goal is to define problem classes for optimization algorithms: optimization algorithms reveal the same behavior if applied to problems of the same equivalence class. Simulation models are equivalent, if they belong to the same problem class. The validation of simulation models can be transfered to the question: do the corresponding models belong to the same problem class?

### 3.2 Standard Validation Techniques

The complete validation process requires subjective and objective comparisons of the model to the real system. Subjective comparisons are judgments of experts ('face validity'), whereas objective tests compare data generated by the model to data generated by the real system. We will discuss objective tests only, for subjective tests, the reader is referred to [BINN01]. Building a model that has a high face validity, validating the model assumptions, and comparing the model input-output transformations to corresponding input-output transformations for the real system can be seen as three widely accepted steps of the validation process [NF67]. In the following we will consider input-output transformations only. The model is viewed as the function:

$$f : (X, D) \to Y \tag{4}$$

Thus values of the uncontrollable input parameters $X$ and values of the controllable decision variables (or of the policy) $D$ are mapped to the output measures $Y$.

The model can be run using generated random variates $X_i$ to produce the simulation-generated output measures. E.g. the S-ring model takes a policy and a system configuration and determines the average number of waiting customers in the system using the generated random variates that determine a customer arrival event.

If real system data is available, a statistical test of the null hypothesis can be conducted:

$$H_0 : \; E(Y) = \mu \text{ is tested against } H_1 : \; E(Y) \neq \mu, \tag{5}$$

where $\mu$ denotes the real system response and $Y$ the simulated model response.

### 3.3 Algorithm Based Validation

Our approach is based on the assumption that specific problems require specific algorithm parameter settings [WM97]. Algorithm based validation (ABV) is related to parameter tuning, but has to be distinguished from parameter control [EHM99]. Parameter control deals with parameter values that are changed during the optimization run, whereas parameter tuning refers to exogenous algorithm parameters to be selected before the optimization run is started [BS02]. Including the optimization algorithm into the validation process, we propose the following methodology [FL01]:

**I. Tuning:** First, the parameter setting of an optimization algorithm for one specific problem is tuned [Bei03]. The S-ring simulation output is the expected average number of floors with waiting customers (minimization problem). Therefore, we can define the performance $Q$ of a policy $\pi$ for a given S-ring setting $S := (n, m, p) \in \mathcal{S}$ as defined in Eq. 2, where $\mathcal{S}$ is the set of all possible S-ring configurations. Furthermore, the following variable $P \in \mathcal{P}$ provides a very compact description of an ES parameter design:

$$P := (\mu_{\text{pop}}, \nu, \kappa, n_\sigma, \tau_0, \tau_i, \rho, R_1, R_2, r_0, N_{\text{tot}}), \tag{6}$$

where $\mu_{\text{pop}}$ denotes the population size, $\nu$ is the offspring–parent ratio, $r_0$ is a random seed, etc. A comprehensive introduction to ES is given in [BS02], whereas [BEM03] and [Bei03] describe the ES-parameterization in detail. $\mathcal{P}$ is the class of all possible ES parameter settings. Summarizing, we have the following function (cf. Eq. 4)

$$g: \ (X, P, S, Q) \to Y \tag{7}$$

where $g$ is defined in terms of the parameter vector $P$, giving the performance of different ES-parameterizations for one pre-specified S-ring model and for one constant quality function $Q$ from the set $\mathcal{Q}$ of quality functions. Therefore, we are able to obtain the expected performance $E(Y)$ of an ES-algorithm for a given problem $S \in \mathcal{S}$ as:

$$E(Y) = g_{S,Q}(X, P), \tag{8}$$

Based on regression analysis, the functional relationship between the parameter settings of algorithms and their expected performance can be specified as a linear model [Kle87, KG92]:

$$E(Y) = X\beta \tag{9}$$

where $Y$ is the vector of ES performance values, $X$ is a matrix of explanatory variables, and $\vec{\beta}$ is the vector of regression parameters.

Recent publications propose generalized linear models (GLMs) [MN89, FL01, Dob02]. A GLM consists of response variables $Y_i$, a parameter vector $\beta$, a set of explanatory (independent or predictor) variables $X$ and a monotone link function $h$ such that

$$h(\mu) = X\beta, \tag{10}$$

where $\mu = E(Y)$. After selecting an adequate family of distributions, the GLM can be fitted.

Finally, optimal algorithm parameter setting $P^*$ can be determined [KG92, MN89].

**II. Extending the single problem to a problem class:** The lift problem can be seen as an extension of the S-ring problem. We assume that both problems belong to the class of ESGC problems $\mathcal{L}$ (validation of the lift model is omitted here, this has been done by Fujitec). To verify our assumption, we extend the model specified in Eq. 10 (or Eq. 9) by introducing a new variable that specifies the underlying optimization problem $L \in \mathcal{L}$ and its corresponding performance measure.

$$h(\mu) = X\beta + \alpha_L + \tilde{X}\gamma_L, \tag{11}$$

The intercept $\alpha_L$ models different algorithm performances. Important in our context are possible interactions between the problem and the model parameters. Scaling of possibly different performance values is not necessary. If there are no interactions, we conclude that the problem is a member of the corresponding class. The inverse handling capacity $F$ as defined in Eq. 1 was used as a corresponding performance measure $Q'$ for the lift model.

Following this approach, we are able to identify problem classes $\mathcal{S}$ by performing a statistical test: The null hypothesis

$$H_0 : \vec{\gamma}_L = 0,\ (L \in \mathcal{S}) \text{ is tested against the alternative } H_1 : \vec{\gamma}_L \neq 0, \tag{12}$$

The test given in Eq. 12 can be regarded as an extension of the standard validation test in Eq. 5.

Finally, we are able to answer the question whether the S-ring model and the lift model belong the same reference class: a pre-specified optimization algorithm shows a similar performance on both problem instances.

# 4 Summary and Outlook

We extended the classical validation approach for simulation models to an algorithm based validation (ABV) approach. A small set of simulation runs can be performed to tune the exogenous parameters of the optimization algorithm. The tuned algorithm can be used to perform the real optimization runs. The S-ring model as a simplified ESGC model was used as a comprehensive example to demonstrate the applicability of this approach. (The full paper will contain numerical examples). ABV can be extended in many ways, e.g. to test the hypothesis that a new operator for an optimization algorithm improves its performance.

# References

[Bar86] G. Barney. *Elevator Traffic Analysis, Design and Control*. Cambridg U.P., 1986.

[Bei03] T. Beielstein. Tuning evolutionary algorithms. Technical Report 148/03, Universität Dortmund, 2003.

[BEM03] T. Beielstein, C.P. Ewald, and S. Markon. Optimal elevator group control by evolution strategies. In *Proc. 2003 Genetic and Evolutionary Computation Conf. (GECCO'03), Chicago*, Berlin, 2003. Springer.

[Bey01] H.-G. Beyer. *The Theory of Evolution Strategies*. Natural Computing Series. Springer, Heidelberg, 2001.

[BFM00] Th. Bäck, D. B. Fogel, and Z. Michalewicz, editors. *Evolutionary Computation 1 – Basic Algorithms and Operators*. Institute of Physics Publ., Bristol, 2000.

[BINN01] J. Banks, J. S. Carson II, B. L. Nelson, and D. M. Nicol. *Discrete Event System Simulation*. Prentice Hall, 2001.

[BS02] H.-G. Beyer and H.-P. Schwefel. Evolution strategies – A comprehensive introduction. *Natural Computing*, 1:3–52, 2002.

[CB98] R.H. Crites and A.G. Barto. Elevator group control using multiple reinforcement learning agents. *Machine Learning*, 33(2-3):235–262, 1998.

[CD98] B. Chopard and M. Droz. *Cellular Automata Modeling of Physical Systems.* Cambridge U.P., 1998.

[Dob02] A.J. Dobson. *An Introduction to Generalized Linear Models.* Chapman and Hall, 2nd edition, 2002.

[EHM99] A.E. Eiben, R. Hinterding, and Z. Michalewicz. Parameter control in evolutionary algorithms. *IEEE Trans. on Evolutionary Computation*, 3(2):124–141, 1999.

[FL01] O. Francois and C. Lavergne. Design of evolutionary algorithms – a statistical perspective. *IEEE Transactions on Evolutionary Computation*, 5(2):129–148, April 2001.

[Gou95] H. Gould. *Thermal and Statistical Physics Simulations.* John Wiley, 1995.

[KG92] J. .P. C. Kleijnen and W. Van Groenendaal. *Simulation - A Statistical Perspective.* Wiley, Chichester, 1992.

[Kle87] J. Kleijnen. *Statistical Tools for Simulation Practitioners.* Marcel Dekker, New York, 1987.

[LK00] A.M. Law and W.D. Kelton. *Simulation Modelling and Analysis.* McGraw-Hill Series in Industrial Egineering and Management Science. McGraw-Hill, New York, 3rd edition, 2000.

[MAB+01] S. Markon, D.V. Arnold, T. Bäck, T. Beielstein, and H.-G. Beyer. Thresholding – a selection operator for noisy ES. In J.-H. Kim, B.-T. Zhang, G. Fogel, and I. Kuscu, editors, *Proc. 2001 Congress on Evolutionary Computation (CEC'01)*, pages 465–472, Seoul, Korea, May 27–30, 2001. IEEE Press, Piscataway NJ.

[Mar95] S. Markon. *Studies on Applications of Neural Networks in the Elevator System.* PhD thesis, Kyoto University, 1995.

[MN89] P. McCullagh and J.A. Nelder. *Generalized Linear Models.* Chapman and Hall, 2nd edition, 1989.

[MN02] S. Markon and Y. Nishikawa. On the analysis and optimization of dynamic cellular automata with application to elevator control. In *The 10th Japanese-German Seminar, Nonlinear Problems in Dynamical Systems, Theory and Applications.* Noto Royal Hotel, Hakui, Ishikawa, Japan, September 2002.

[NF67] T.H. Naylor and J.M. Finger. Verification of computer simulation models. *Management Science*, 2:B92–B101, 1967.

[SC99] A.T. So and W.L. Chan. *Intelligent Building Systems.* Kluwer A.P., 1999.

[SWW02] H.-P. Schwefel, I. Wegener, and K. Weinert, editors. *Advances in Computational Intelligence – Theory and Practice.* Natural Computing Series. Springer, Berlin, 2002.

[WM97] D.H. Wolpert and W.G. Macready. No free lunch theorems for optimization. *IEEE Transactions on Evolutionary Computation*, 1(1):67–82, 1997.