Algorithm Based Validation of a Simplified Elevator Group Controller Model

Thomas Bartz–Beielstein* Sandor Markon† Mike Preuss*

*Universerstität 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

Today's urban life cannot be imagined without elevators. 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]. It 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 currently 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 constructed as a simplified model of an ESGC system using 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 not trivial, 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 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. Tuning the optimization algorithm on the simplified simulation model results in a good parameter setting for the optimization algorithm, that is 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 elements of this algorithm based validation method are shown in Fig. 2.

This paper is organized as follows: in Section 2, the ESGC problem is introduced. Section 3 discusses different validation techniques. A concrete numerical example is presented in Sec. 4. Section 5 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 traditional elevators customers can only press a button to request up or down service. They choose the exact destination after entering the elevator car. In a destination call system, the desired destination can be chosen at a terminal before entering the elevator car [BEM03]. Fujitec, one of the world's leading elevator manufacturers developed a controller that is trained by use of a set of fuzzy controllers. Each controller represents control strategies for different traffic situations [Mar95]. The NN structure and the neural weights determine a concrete control strategy. The network structure as well as many of the weights remain fixed, only some of the weights on the output layer can be modified and optimized. A discrete-event based elevator group simulator is used to compute the controller's performance. This ESGC simulation model will be referred to as the 'lift model' (or simply 'lift') throughout the rest of this paper.

The identification of globally optimal weights is a highly complex task since the topology of the fitness function is highly non-linear and highly multi-modal. It is stochastically disturbed due to the nondeterminism of service calls, and dynamically changing with respect to traffic loads. Gradient based optimization techniques cannot be applied successfully to this optimization problem. The computational effort for single simulator runs limits the maximum number of fitness function evaluations to the order of magnitude 10^4.

2.2 The Lift Model as an Optimization Problem

The objective function considered here is the average waiting time of all passengers served during a simulated elevator movement. Different traffic patterns occur during this simulation:

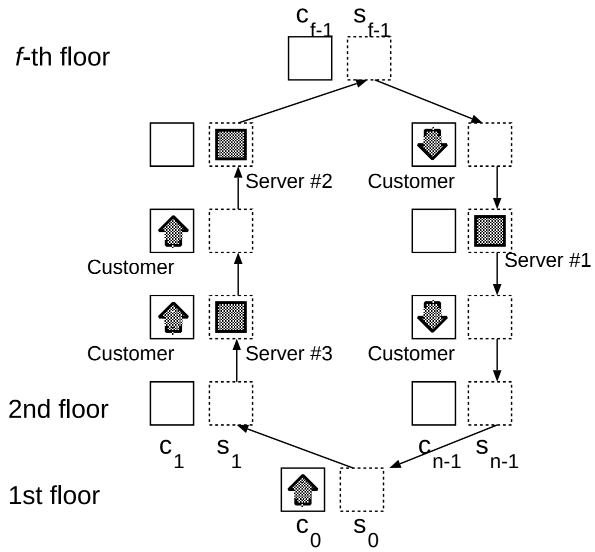

Figure 1: The S-ring as an elevator system. The states are numbered from 0 to $n-1$.

Up-peak traffic during the morning rush-hour, less intense, balanced traffic during the day (two-way traffic) and rush hour traffic at closing time (down-peak traffic).

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. 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. The handling capacities were averaged and then subtracted from 3000 pass./h to obtain a minimization problem. The latter value was empirically chosen as an upper bound for the given scenario. Eq. 1 shows the resulting fitness function.

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

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. $F(x)$ is called the 'inverse handling capacity' in the following.

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. Although ESGC models and the S-ring show similar dynamics, there are some major differences that 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]. There are m elevator cars among n sites, and a new customer arrives on the ith floor with probability p. A 2-bit state (s_i, c_i) is associated with each site. The

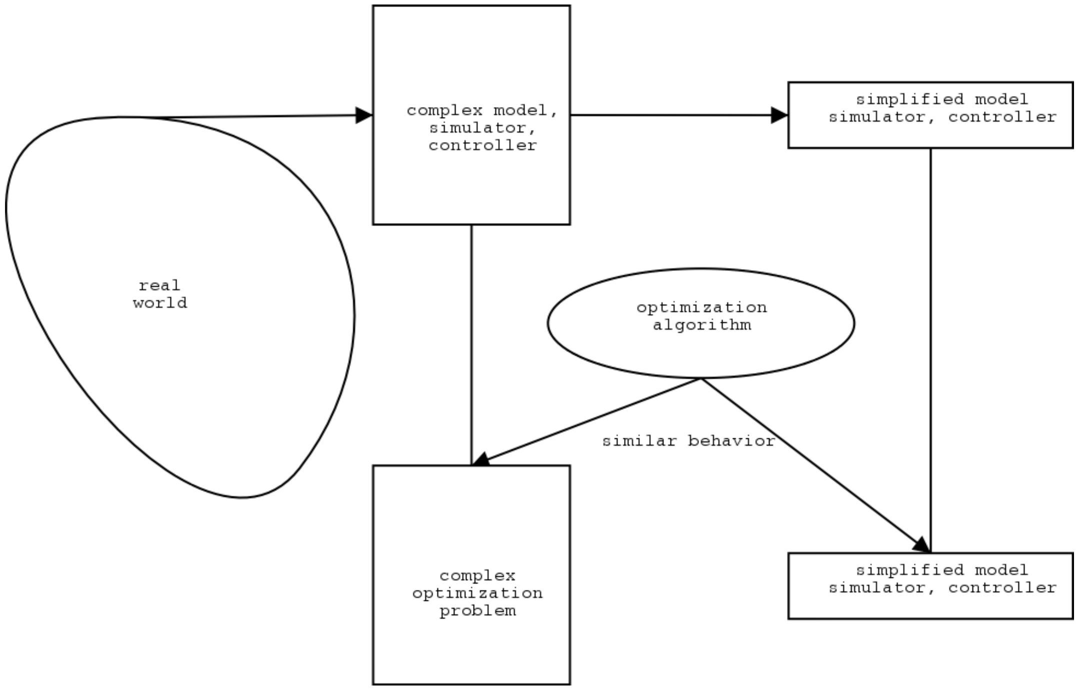

Figure 2: Elements of algorithm based validation. The simplification of the complex model is valid if the optimization algorithm reveals a similar behavior on the corresponding complex optimization problem and the corresponding simplified problem.

following vector describes the state at time t of the S-ring:

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

The s_i bit is set to 1 if a server is present on the ith 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 ith floor. The sites are not updated synchronously, an 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' $\pi : \mathbf{X} \to \{0,1\}$.

2.4 The S-Ring Model as an Optimization Problem

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 π, 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, referred to in the following as the S-ring optimization problem (S-ring problem), is to find π^* for given parameters n, m, and p. The exact performance of a particular policy cannot be determined, it must be estimated. Problems related to optimization via simulation are discussed in [BINN01].

3 The S-Ring Model as a Valid ESGC Model

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. Let the random variable Y denote the result of a simulation run or some measure of system performance in stochastic simulation. The goal of 'optimization via simulation' is to minimize or maximize its expectation $E(Y)$[BINN01]. We consider optimization algorithms to define equivalence classes for optimization problems: an optimization algorithm reveals the same behavior if applied to problems of the same equivalence class. Since the behavior of an optimization algorithm can be determined by running the simulation, we can state: Simulation models are equivalent, if their corresponding optimization problems belong to the same equivalence class. Therefore we will distinguish models (e.g. the S-ring model) and their related problems (e.g. the S-ring problem). The validation of simulation models can be transfered to the question: are their corresponding optimization problems equivalent?

3.1 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

Table 1: DoE parameter setting for ES

Symbol	Parameter	Typical Values	Low	Medium	High
μ	number of parent individuals	$10 \ldots 100$	5	12	19
$\nu = \lambda/\mu$	offspring-parent ratio	$1 \ldots 10$	2	4	6
n_σ	number of standard deviations	$1 \ldots D$	-	1	-
τ_0	global mutation parameter	$1/\sqrt{2\sqrt{D}}$	-	$1/\sqrt{2\sqrt{D}}$	-
τ_i	individual mutation parameter	$1/\sqrt{2D}$	-	$1/\sqrt{2D}$	-
κ	age	$1 \ldots \infty$	1		∞
ρ	mixing number	μ	-	2	-
R_1	recombination operator for object variables	{discrete}	-	{discrete}	-
R_2	recombination operator for strategy variables	{intermediate}	-	{intermediate}	-

and determines the expected average number of floors with 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 μ denotes the real system response and Y the simulated model response.

3.2 Algorithm Based Validation

Our goal is to show that the S-ring model is a valid ESGC-model. The first step in the algorithm based validation (ABV) approach requires the selection of an optimization algorithm. Evolution strategies (ES), that are applicable to many optimization problems, have been chosen in the following [BS02]. 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: The parameter setting of an optimization algorithm for one specific problem can be tuned in the following manner [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 π 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 μ_{pop} denotes the population size, ν is the offspring–parent ratio, r_0 is a random seed, etc. These parameters, that are kept constant during the optimization run, are called 'exogenous strategy parameters'. A comprehensive introduction to ES is given in [BS02], whereas [BEM03] and [Bei03] describe the ES-parameterization in detail. Let $\mathcal{P}$ denote 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 β is the vector of regression parameters.

Recent publications propose generalized linear models (GLMs) [MN89, FL01, LMM01, Dob02]. GLM analysis provides an unified approach for linear and non-linear models with both normal and non-normal responses. A GLM consists of response variables Y_i, a parameter vector β, a set of explanatory (independent or predictor) variables X and a monotone link function h such that

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

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

Finally, the 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 $\mathcal{L}$. The validation of the lift model is omitted here. A detailed validation of the lift model has been done by the elevator manufacturer.

There are at least two different approaches to handle different ranges of the response values: the success-rate approach fits a generalized linear model to binomial data and enables the comparison of different response values without rescaling, whereas the factor approach introduces an extra factor with two levels to model the influence of the problem on the ES performance.

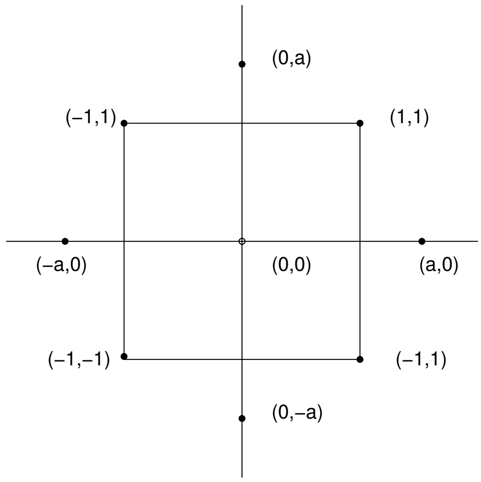

Figure 3: Central composite design

In the success-rate approach the algorithms performance is measured by a binary variable: it takes the value 1 if the run was successful, otherwise it takes the value 0 [BDFP02]. As n repeat runs are performed for each run configuration (cf. Eq.6), the proportion of successful runs can be determined. Based on the success rate, we can apply logistic regression to analyze the algorithms behavior [MN89, Col91, Dob02].

In the factor approach the model specified in Eq. 10 (or Eq. 9) is extended by introducing a new variable that specifies the underlying optimization problem $L \in \mathcal{L}$ and its corresponding performance measure to test the algorithms for similar performance on different problems:

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

$\tilde{X}$ denotes the modified matrix of explanatory variables. The intercept α_L models different algorithm performances, therefore scaling of possibly different performance values is not necessary. Important in our context are possible interactions ($\tilde{X}\gamma_L$) between the problem and the model parameters. 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 a problem class $\mathcal{L}$ by performing a statistical test: The null hypothesis

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

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 to the same reference class: a pre-specified optimization algorithm shows a similar performance on both problem instances if this is true.

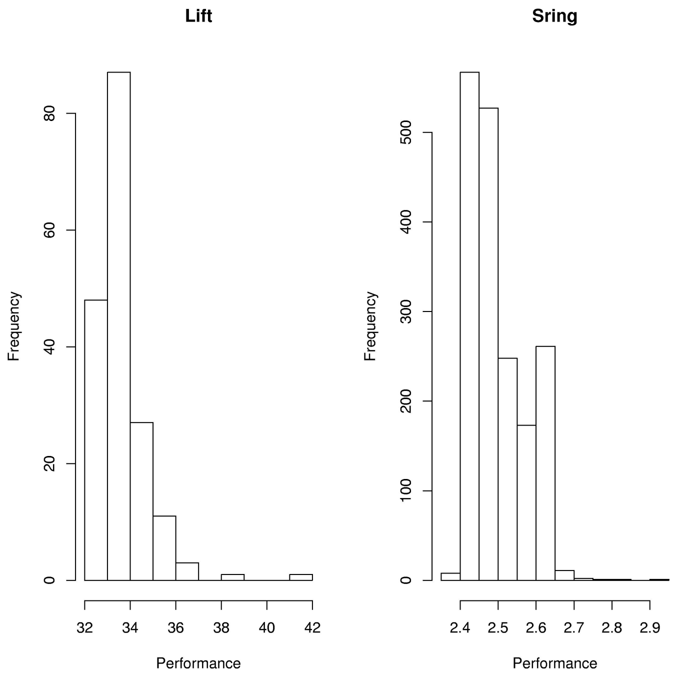

Figure 4: Comparison of the lift and S-ring performance.

4 Example

Since ABV extends techniques that enable the determination of an improved parameter setting for one specific algorithm-problem pair (algorithm tuning), this technique is described first. Therefore, the first objective is to find an optimal setting for the exogenous strategy parameter of an ES for one specific problem.

4.1 Algorithm Tuning

1. The first step is to select a distribution of the performance measure for the optimization problem. This can be done by using frequency tables, graphical methods such as histograms, or statistical tests (i.e. Kolmogorov-Smirnov, Shapiro-Wilk). Plotting the quantiles of the performance samples against those of the reference distribution can be used to compare different distribution hypotheses (i.e. the Gaussian and the Gamma hypothesis).

2. We recommend to start with the canonical link function, since its adequacy can be checked after other model issues have been considered.

3. The model search (determination of the predictors, their orders and interactions) can be based on the Akaike Information Criterion (AIC). Backward elimination starts with an over-fitted model, whereas forward selection adds a new variable to the model at each stage of the process.

4. From the final model we get estimates of the regression coefficients that are used to determine an improved strategy parameter setting P^*.

A numerical example will be given in the next subsection to demonstrate how parameter tuning can be extended to study equivalence classes.

Table 2: Output generated in R for the GLM, Sec. 4.2

	Estimate	Std. Error	t value	Pr(>|t|)
(Intercept)	0.4014	0.0017	231.75	0.0000
Lift	−0.3715	0.0011	−353.50	0.0000
P:poly(R, 2)1	−0.0002	0.0019	−0.10	0.9202
P:poly(R, 2)2	0.0009	0.0010	0.86	0.3891
R:K-1	−0.0022	0.0004	−5.62	0.0000
R:K1	−0.0000	0.0004	−0.01	0.9889
R:poly(P, 2)1	−0.0009	0.0035	−0.26	0.7967
R:poly(P, 2)2	−0.0109	0.0024	−4.61	0.0000
poly(R, 2)1:K1	0.0157	0.0219	0.72	0.4735
poly(R, 2)2:K1	−0.0191	0.0191	−1.00	0.3183
P:R:K-1	0.0001	0.0000	7.11	0.0000

4.2 Equivalence Classes for Problems

A central composite design (CCD) depictured in Fig. 3 was used to set up the experiments. The corresponding ES parameter settings are shown in Tab. 1. Histograms of the best fitness value obtained during an optimization run can be used as a starting point to answer the question whether two problems L_1 and L_2 are in the same equivalence class. The histograms of the performance function for the lift and the S-ring problem are shown in Fig. 4. Both histograms show assymetric empirical distributions that appear similar. Therefore a generalized linear model that is based on the Gamma distribution and the canonical link is used in the following analysis. A factor L with two levels {Sring, Lift} is introduced to check for interactions between algorithm parameters and L. The model algebra introduced by Wilkinson and Rogers will be used to describe the regression model [WR73]: `A+B+A:B` reads 'effect of factor `A` plus effect of factor `B` plus interaction between `A` and `B`' and can be abbreviated as `A*B`. Starting from the overfitted model with three factor interactions we perform a backward elimination procedure. The function `stepAIC` provided by the statistical software package R was used to automate the first steps in the process of model selection. R's `dropterm` function was used to select significant terms manually. If `P` denotes the number of parent individuals, `R` the parent-offspring ratio, and `K` the age, the final model reads

$$\mathtt{Y} \sim \mathtt{Function} + \mathtt{P} : \mathtt{poly}(\mathtt{R}, 2) + \mathtt{R} : \mathtt{K} + \mathtt{R} : \mathtt{poly}(\mathtt{P}, 2) + \mathtt{K} : \mathtt{poly}(\mathtt{R}, 2) + \mathtt{P} : \mathtt{R} : \mathtt{K}, \quad (13)$$

and will be written in the following as `Y` $\sim$ `Function+ model`. The model coefficients, that can be used for algorithm tuning, are shown in Tab. 2. The last column in Tab. 2 shows that there are no significant interactions between the problem L and other factors.

Another way to test for interactions between the factor L and other factors is to compare two nested models $M_2 \subset M_1$. M_1 includes interactions between L and other factors of the reduced model in Eq. 13, whereas interactions are omitted in M_2. M_1: `Y ~ Function * model` is compared to M_2: `Y ~ Function + model.` The symbol '`+`' denotes independent factors and the symbol '`*`' denotes interactions between two factors. This procedure leads to the result shown in Tab. 3. This ANOVA table indicates that there is no significant difference if interactions between L and the other factors are included. We can conclude that the

Table 3: ANOVA

	Resid. Df	Resid. Dev	Df	Deviance	F	Pr(>F)
1	1960	10.36				
2	1969	10.47	−9	−0.11	1.02	0.4229

S-ring problem and the lift problem belong to the same problem class. Therefore the S-ring model can be seen as a valid simplification of the lift model.

5 Summary and Outlook

We extended the classical validation approach for simulation models to an algorithm based validation (ABV) approach. A simulation run S equipped with an objective function Q defines an optimization problem (S, Q). If an optimization algorithm reveals the same behavior on two different optimization problems (S, Q) and (S', Q'), the corresponding simulation models S and S' are considered as equivalent. The statistical methodology introduced by Fr̨ancois and Lavergne was used to specify the equivalence of optimization problems [FL01]. Problem equivalence was used to check the model validity.

Thus the optimization practitioner is able to perform simulation runs on the simplified problem 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. Subject of our current research is the analysis of different S-ring problem classes. These S-ring classes can represent different lift configurations and traffic patterns.

Alternatively to the approach presented here, the algorithm performance can be measured by a binary variable Y_i. If the optimization run was successful (e.g. if the algorithm has located an optimum), Y_i takes the value 1 (success). Otherwise Y_i is set to 0 (failure). Since each optimization run is repeated several times with different random seeds, the result of an experiment can be described as the quotient successes/failures. A logistic regression model can be used to perform an analysis that is similar to the approach presented in this paper.

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.

[BDFP02] T. Beielstein, J. Dienstuhl, C. Feist, and M. Pompl. Circuit design using evolutionary algorithms. In David B. Fogel, Mohamed A. El-Sharkawi, Xin Yao, Garry Greenwood, Hitoshi Iba, Paul Marrow, and Mark Shackleton, editors, *Proceedings of the 2002 Congress on Evolutionary Computation CEC2002*, pages 1904–1909. IEEE Press, 2002.

[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.

[Col91] D. Collett. *Modelling Binary Data.* Chapman and Hall, 1991.

[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.

[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.

[LMM01] S.L. Lewis, D.C. Montgomery, and R.H. Myers. Examples of designed experiments with nonnormal responses. *Journal of Quality Technology*, 33(3):265–278, July 2001.

[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.* unknown, 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.

[WR73] G. N. Wilkinson and C. E. Rogers. Symbolic description of factorial models for analysis of variance. 22(3):392–399, 1973.