Optimal Elevator Group Control by Evolution Strategies

Thomas Beielstein[1], Claus-Peter Ewald[2], and Sandor Markon[3]

[1] Universtität Dortmund, D-44221 Dortmund, Germany
`Thomas.Beielstein@udo.edu`,
WWW home page: `http://ls11-www.cs.uni-dortmund.de/people/tom/index.html`
[2] NuTech Solutions GmbH, Martin-Schmeisser Weg 15, D-44227 Dortmund, Germany
`Ewald@nutechsolutions.de`,
WWW home page: `http://www.nutechsolutions.de`
[3] FUJITEC Co.Ltd. World Headquarters, 28-10, Shoh 1-chome, Osaka, Japan
`markon@rd.fujitec.co.jp`,
WWW home page: `http://www.fujitec.com`

Abstract. Efficient elevator group control is important for the operation of large buildings. Recent developments in this field include the use of fuzzy logic and neural networks. This paper summarizes the development of an evolution strategy (ES) that is capable of optimizing the neuro-controller of an elevator group controller. It extends the results that were based on a simplified elevator group controller simulator. A threshold selection technique is presented as a method to cope with noisy fitness function values during the optimization run. Experimental design techniques are used to analyze first experimental results.

1 Introduction

Elevators play an important role in today's urban life. The elevator supervisory group control (ESGC) problem is related to many other stochastic traffic control problems, especially with respect to the complex behavior and to many difficulties in analysis, design, simulation, and control [Bar86,MN02]. The ESGC problem has been studied for a long time: first approaches were mainly based on analytical approaches derived from queuing theory, in the last decades artificial intelligence techniques such as fuzzy logic (FL), neural networks (NN), and evolutionary algorithms (EA) were introduced, whereas today hybrid techniques, that combine the best methods from the different worlds, enable improvements. Therefore, the present era of optimization could be classified as the era of computational intelligence (CI) methods [SWW02,BPM03]. CI techniques might be useful as quick development techniques to create a new generation of self-adaptive ESGC systems that can handle high maximum traffic situations.

In the following we will consider an ESGC system that is based on a neural network 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. Let $\boldsymbol{x}$ denote one weight configuration. We can define the optimal weight configuration as $\boldsymbol{x}^* = \arg\min f(\boldsymbol{x})$, where the performance measure $f()$ to be minimized is defined later. The determination of an optimal weight setting $\boldsymbol{x}^*$ is difficult, since it is not trivial

1. to find an efficient strategy that modifies the weights without generating too many infeasible solutions, and
2. to judge the performance or fitness $f(\boldsymbol{x})$ of one ESGC configuration. The performance of one specific weight setting $\boldsymbol{x}$ is based on simulations of specific traffic situations, which lead automatically to stochastically disturbed (noisy) fitness function values $\tilde{f}(\boldsymbol{x})$.

The rest of this article deals mainly with the second problem, especially with problems related to the comparison of two noisy fitness function values. Before we discuss a technique that might be able to improve the comparison of stochastically disturbed values in section 3, we introduce one concrete variant of the ESGC problem in the next section. The applicability of the comparison technique to the ESGC problem is demonstrated in section 4, whereas section 5 gives a summary and an outlook.

2 The Elevator Supervisory Group Control Problem

In this section, we will consider the elevator supervisory group control problem [Bar86,SC99,MN02]. An elevator group controller assigns elevators to service calls. An optimal control strategy is a precondition to minimize service times and to maximize the elevator group capacity. Depending on current traffic loads, several heuristics for reasonable control strategies do exist, but which in general lead to suboptimal controllers. These heuristics have been implemented by Fujitec, one of the world's leading elevator manufacturers, using a fuzzy control approach. In order to improve the generalization of the resulting controllers, Fujitec developed a neural network based controller, which is trained by use of a set of the aforementioned fuzzy controllers, each representing control strategies for different traffic situations. This leads to robust and reasonable, but not necessarily optimal, control strategies [Mar95].

Here we will be concerned with finding optimal control strategies for destination call based ESGC systems. 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 lets the customer choose the desired destination at a terminal before entering the elevator car. This provides more exact information to the group controller, and allows higher efficiency by grouping of passengers into elevators according to their destinations; but also limits the freedom of decision. Once a customer is assigned to a car and the car number appears on the terminal, the customer moves away from the terminal, which makes it very inconvenient to reassign his call to another car later.

The concrete control strategy of the neural network is determined by the network structure and neural weights. While the network structure as well as

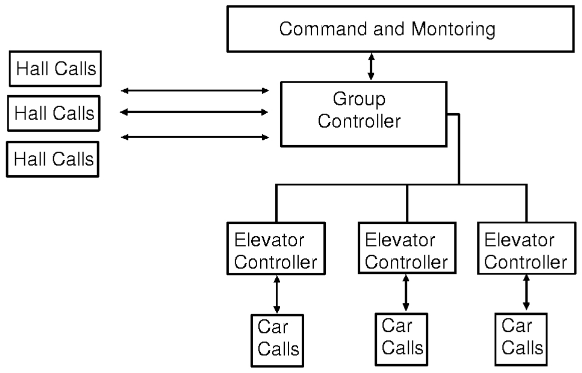

Fig. 1. The architecture of an elevator supervisory group control system [Sii97].

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, an algorithm is searched for 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. Fig.2 illustrates how the output from the simulator is visualized.

Unfortunately, the resulting response surface shows some characteristics which makes the identification of globally optimal weights difficult if not impossible. The topology of the fitness function can be characterized as follows:

– highly nonlinear,
– highly multi-modal,
– varying fractal dimensions depending on the position inside the search space,
– randomly disturbed due to the nondeterminism of service calls,
– dynamically changing with respect to traffic loads,
– 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. Consequently, efficient robust methods from the domain of black-box optimization are required where evolutionary computation is one of the most promising approaches. Therefore, we have chosen an evolution strategy to perform the optimization [BFM00,Bey01].

The objective function for this study is the average waiting time of all passengers served during a simulated elevator movement of two hours. Further experiments will be performed in future to compare this objective function to more complex ones, which e.g. take into account the maximum waiting time as well. There are three main traffic patterns occurring during a day: up-peak (morning

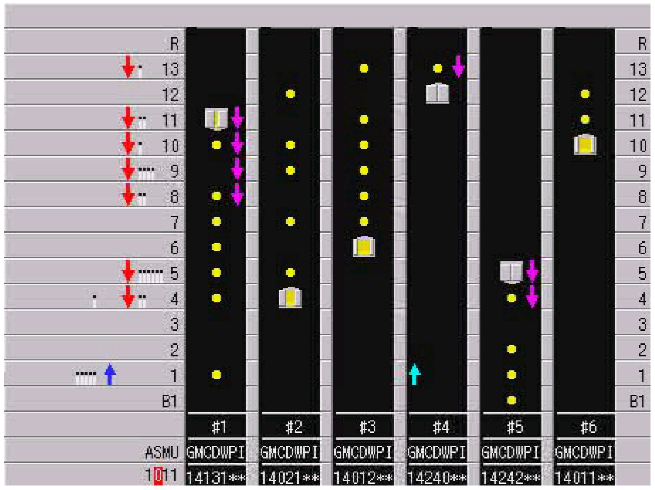

Fig. 2. Visualization of the output from the elevator group simulator.

rush hour), two-way (less intense, balanced traffic during the day), and down-peak traffic (rush hour at closing time). These patterns make up the simulation time to one third each, which forces the resulting controller to cope with different traffic situations. As described above, the objective function values are massively disturbed by noise, since the simulator calculates different waiting times on the same objective variables (network weights), which stems from changing passenger distributions generated by the simulator. For the comparison of different ES parameter settings the final 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 simulator with altering random seeds and increasing passenger loads using the network weights of the best individuals found in each optimization run. Finally, to enable the deployment of our standard evaluation process, we needed a single figure to minimize. Therefore, 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 (9) and is called 'inverse handling capacity' in the following.

3 Evolution Strategies and Threshold Selection

3.1 Experimental Noise and Evolution Strategies

In this section we discuss the problem of comparing two solutions, when the available information (the measured data) is disturbed by noise. A bad solution might appear better than a good one due to the noise. Since minor differences are unimportant in this situation, it might be useful to select only much better values. A threshold value τ can be used to formulate the linguistic expression 'much better' mathematically:

$$x \text{ is much better (here: greater) than } y \Leftrightarrow \; \tilde{f}(x) > \tilde{f}(y) + \tau. \tag{1}$$

Since comparisons play an important role in the selection process of evolutionary algorithms, we will use the term threshold selection (TS) in the rest of this article. TS can be generalized to many other stochastic search techniques such as particle swarm optimizers or genetic algorithms. We will consider evolution strategies in this article only. The goal of the ES is to find a vector $\boldsymbol{x}^*$, for which holds:

$$f(\boldsymbol{x}^*) \leq f(\boldsymbol{x}) \quad \forall \boldsymbol{x} \in \mathcal{D}, \tag{2}$$

where the vector $\boldsymbol{x}$ represents a set of (object) parameters, and $\mathcal{D}$ is some n-dimensional search space. An ES individual is usually defined as the set of object parameters $\boldsymbol{x}$, strategy parameters $\boldsymbol{s}$, and its fitness value $f(\boldsymbol{x})$ [BS02].

In the following we will consider fitness function values obtained from computer simulation experiments: in this case, the value of the fitness function depends on the seed that is used to initialize the random stream of the simulator. The exact fitness function value $f(\boldsymbol{x})$ is replaced by the noisy fitness function value $\tilde{f}(\boldsymbol{x}) = f(\boldsymbol{x}) + \epsilon$. In the theoretical analysis we assume normal-distributed noise, that is $\epsilon \sim \mathcal{N}(\mu, \sigma^2)$. It is crucial for any stochastic search algorithm to decide with a certain amount of confidence whether one individual has a better fitness function value than its competitor. This problem will be referred to as the selection problem in the following [Rud98,AB01]. Reevaluation of the fitness function can increase this confidence, but this simple averaging technique is not applicable to many real-world optimization problems, i.e. if the evaluations are too costly.

3.2 Statistical Hypothesis Testing

Threshold selection belongs to a class of statistical methods that can reduce the number of reevaluations. It is directly connected to classical statistical hypothesis testing [BM02].

Let $\boldsymbol{x}$ and $\boldsymbol{y}$ denote two object parameter vectors, with $\boldsymbol{y}$ being proposed as a 'better' (greater) one, to replace the existing $\boldsymbol{x}$. Using statistical testing, the selection problem can be stated as testing the null hypothesis $H_0 : f(\boldsymbol{x}) \leq f(\boldsymbol{y})$ against the one-sided alternative hypothesis $H_1 : f(\boldsymbol{x}) > f(\boldsymbol{y})$. Whether the null hypothesis is accepted or not depends on the specification of a critical region for the test and on the definition of an appropriate test statistic. Two kinds of errors may occur in testing hypothesis: if the null hypothesis is rejected when it is true, an alpha error has been made. If the null hypothesis is not rejected when it is false, a beta error has been made. Consider a maximization problem: the *threshold rejection probability* P_τ^- is defined as the conditional probability,

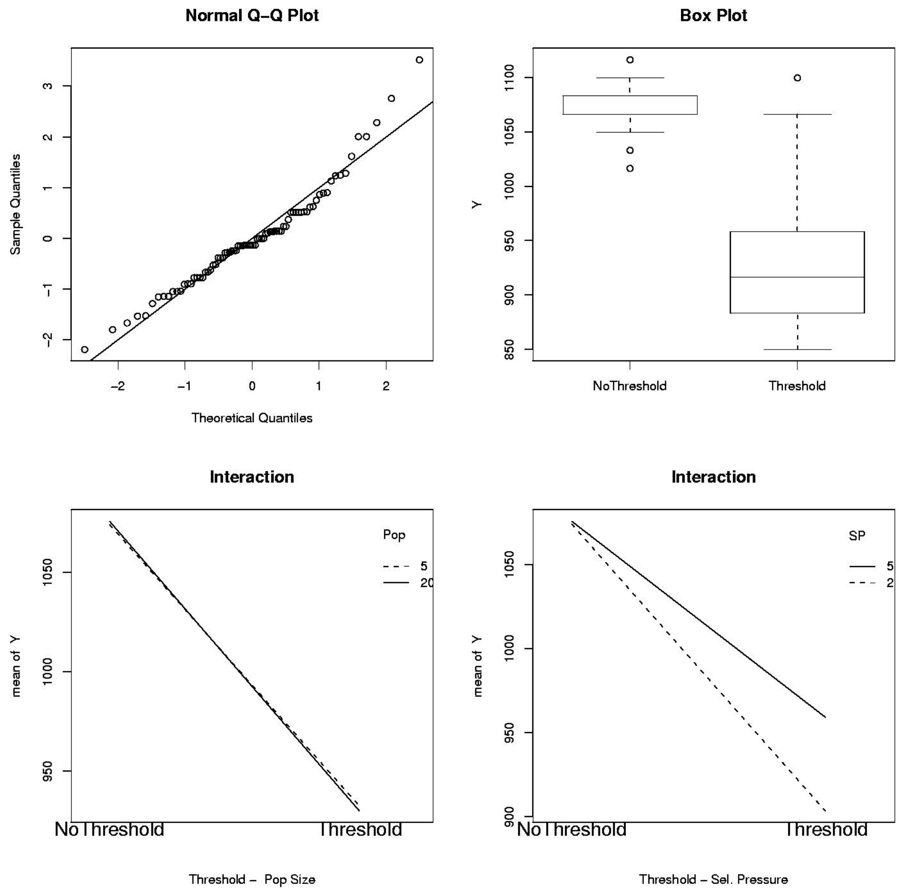

Fig. 3. Hypothesis testing to compare models: we are testing whether threshold selection has any significance in the model. The Normal QQ-plot, the box-plot, and the interaction-plots lead to the conclusion that threshold selection has a significant effect. Y denotes the fitness function values that are based on the inverse handling capacities: smaller values are better, see (9). Threshold, population size μ and selective pressure (offspring-parent ratio ν) are modified according to the values in Tab. 3. The function $\alpha(t)$ as defined in (6) was used to determine the threshold value.

that a worse candidate $\boldsymbol{y}$ has a better noisy fitness value than the fitness value of candidate $\boldsymbol{x}$ by at least a threshold τ:

$$P_\tau^- := P\{\overline{f}(\boldsymbol{y}) \leq \overline{f}(\boldsymbol{x}) + \tau \mid f(\boldsymbol{y}) \leq f(\boldsymbol{x})\}, \tag{3}$$

where $\overline{f}(\boldsymbol{x}) := \sum_{i=1}^{n} \tilde{f}(\boldsymbol{x}_i)/n$ denotes the sample average of the noisy values [BM02]. P_τ^- and the alpha error are complementary probabilities:

$$P_\tau^- = 1 - \alpha. \tag{4}$$

(4) reveals, how TS and hypothesis testing are connected: maximization of the threshold rejection probability, an important task in ES based optimization, is equivalent to the minimization of the alpha error. Therefore, we provide techniques from statistical hypothesis testing to improve the behavior of ES in noisy environments.

Our implementation of the TS method is based on the common practice in hypothesis testing to specify a value of the probability of an alpha error, the so called significance level of the test. In the first phase of the search process the explorative character is enforced, whereas in the second phase the main focus lies on the exploitive character. Thus, a technique that is similar to simulated annealing is used to modify the significance level of the test during the search process. In the following paragraph, we will describe the TS implementation in detail.

3.3 Implementation Details

The TS implementation presented here is related to doing a hypothesis test to see whether the measured difference in the expectations of the fitness function values is significantly different from zero. The test result (either a 'reject' or 'fail-to-reject' recommendation) determines the decision whether to reject or accept a new individual during the selection process of an ES. In the following, we give some recommendations for the concrete implementation. We have chosen a parametric approach, although non-parametric approaches might be applicable in this context too.

Let $Y_{i1}, Y_{i2}, \dots Y_{in}$ $(i = 1, 2)$ be a sample of n independent and identically distributed (i.i.d.) measured fitness function values. The n differences $Z_i = Y_{1j} - Y_{2j}$ are also i.i.d. random variables (r.v.) with sample mean $\overline{Z}$ and sample variance $S^2(n)$. Consider the following test on means μ_1 and μ_2 of normal distributions: if $H_0 : \mu_1 - \mu_2 \leq 0$ is tested against $H_1 : \mu_1 - \mu_2 > 0$, we have the test statistic

$$t_0 = \frac{\overline{z} - (\mu_1 - \mu_2)}{s\sqrt{2/n}}, \tag{5}$$

with sample standard deviation S (small letters denote realizations of the corresponding r.v.). The null hypothesis $H_0 : \mu_1 - \mu_2 \leq 0$ would be rejected if $t_0 > t_{2n-2,1-\alpha}$, where $t_0 > t_{2n-2,1-\alpha}$ is the upper α percentage point of the t distribution with $2n - 2$ degrees of freedom.

Summarizing the methods discussed so far, we recommend the following recipe:

1. Select an alpha value. The α error is reduced during the search process, e.g. the function
$$\alpha(t) = \sqrt{(1 - t/t_{\max})}/2, \tag{6}$$
with $t_{\max}$ the maximum number of iterations and $t \in \{0, t_{\max}\}$, the actual iteration number, can been used.
2. Evaluate the parent and the offspring candidate n times. To reduce the computational effort, this sampling can be performed every k generations only.
3. Determine the sample variance.
4. Determine the threshold value $\tau = t_{2n-2,1-\alpha} \cdot s \cdot \sqrt{2/n}$
5. A new individual is accepted, if its (perturbed) fitness value plus τ is smaller than the (perturbed) fitness value of the parent.

A first analysis of threshold selection can be found in [MAB$^+$01,BM02]. In the following section we will present some experimental results that are based on the introduced ideas.

4 Threshold Selection in the Context of ESGC Optimization

4.1 Statistical experimental design

The following experiments have been set up to answer the question: how can TS improve the optimization process if only stochastically perturbed fitness function values (e.g. from simulation runs) can be obtained? Therefore, we simulated alternative ES parameter configurations and examined their results. We wanted to find out if TS has any effect on the performance of an ES, and if there are any interactions between TS and other exogenous parameters such as population size or selective pressure. A description of the experimental design (DoE) methods we used is omitted here. [Kle87,LK00] give excellent introductions into design of experiments, the applicability of DoE to evolutionary algorithms is shown in [Bei03].

The following vector notation provides a very compact description of evolution strategies [BS02,BM02]. Consider the following parameter vector of an ES parameter design:

$$\boldsymbol{p}_{\mathrm{ES}} = (\mu, \nu, S, n_\sigma, \tau_0, \tau_i, \rho, R_1, R_2, r_0)\,, \tag{7}$$

where $\nu := \lambda/\mu$ defines the offspring-parent ratio, $S \in \{C, P\}$ defines the selection scheme resulting in a comma or plus strategy. The representation of the selection parameter S can be generalized by introducing the parameter κ that defines the maximum age (in generations) of an individual. If κ is set to 1, we obtain the comma-strategy, if κ equals $+\infty$, we model the plus-strategy. The mixing number ρ defines the size of the parent family that is chosen from the parent pool of size μ to create λ offsprings. We consider global intermediate GI, global discrete GD, local intermediate LI, and local discrete LD recombination.

R_1 and R_2 define the recombination operator for object resp. strategy variables, and r_0 is the random seed. This representation will be used throughout the rest of this article and is summarized in Tab. 1. Typical settings are:

$$\boldsymbol{p}_{\text{ES}} = \left(5, 7, 1, 1, 1/\sqrt{2\sqrt{D}}, 1/\sqrt{2D}, 5, GD, GI, 0\right). \tag{8}$$

Our experiment with the elevator group simulator involved three factors. A 2^3 full factorial design has been selected to compare the influence of different population sizes, offspring-parent ratios and selection schemes. This experimental design leads to eight different configurations of the factors that are shown in Tab.1. The encoding of the problem parameters is shown in Tab.2. Tab.3 displays the parameter settings that were used during the simulation runs: the population size was set to 5 and 20, whereas the parent-offspring ratio was set to 2 and 5. This gives four different (parent, offspring) combinations: (5,20), (5,25), (20,40), and (20,100). Combined with two different selection schemes we obtain 8 different run configurations. Each run configuration was repeated 10 times.

4.2 Analysis

Although a batch job processing system, that enables a parallel execution of simulation runs, was used to run the experiments, more than a full week of round-the-clock computing was required to perform the experiments. Finally 80 configurations have been tested (10 repeats of 8 different factor settings). The simulated inverse handling capacities can be summarized as follows:

```
Min.   : 850.0   1st Qu.:916.7   Median :1033.3
Mean   :1003.1   3rd Qu.:1083.3  Max.   :1116.7
```

A closer look at the influence of TS on the inverse handling capacities reveals that ES runs with TS have a lower mean (931.2) than simulations that used a plus selection strategy (1075.0). As introduced in section 2, the fitness function reads (minimization task):

$$g(\boldsymbol{x}) = 3000.0 - \overline{f}_{\boldsymbol{p}_{\text{ES}}}(\boldsymbol{x}), \tag{9}$$

where the setting from Tab. 3 was used, $\overline{f}_{\boldsymbol{p}_{\text{ES}}}$ is the averaged handling capacity (pass./h), and $\boldsymbol{x}$ is a 36 dimensional vector that specifies the NN weights. The minimum fitness function value (850.0) has been found by TS. Performing a t-test (the null hypothesis 'the true difference in means is not greater than 0' that is tested against the 'alternative hypothesis: the true difference in means is greater than 0') leads to the p-value smaller than $2.2e-16$.

An interaction plot plots the mean of the fitness function value for two-way combinations of factors, e.g. population size and selective strength. Thus, it can illustrate possible interactions between factors. Considering the interaction plots in Fig. 3, we can conclude that the application of threshold selection improves the

Table 1. DoE parameter for ES.

Symbol	**Parameter**	**Recommended Values**
μ	number of parent individuals	$10 \dots 100$
$\nu = \lambda/\mu$	offspring-parent ratio	$1 \dots 10$
n_σ	number of standard deviations	$1 \dots D$
τ_0	global mutation parameter	$1/\sqrt{2\sqrt{D}}$
τ_i	individual mutation parameter	$1/\sqrt{2D}$
κ	age	$1 \dots \infty$
ρ	mixing number	μ
R_1	recombination operator for object variables	{discrete}
R_2	recombination operator for strategy variables	{intermediate}

Table 2. DoE parameter for the fitness function.

Symbol	**Parameter**	**Values**
f	fitness function, optimization problem	ESGC problem, minimization, see (9)
D	dimension of f	36
N_{exp}	number of experiments for each scenario	10
N_{tot}	total number of fitness function evaluations	$5 \cdot 10^3$
σ	noise level	unknown

Table 3. ES parameter designs.

ES	**ESGC-Design Model** ($D = 36$)	
Variable	Low (-1)	High ($+1$)
μ	5	20
ν	2	5
κ	$+\infty$	Threshold Selection
The following values remain unchanged:		
n_σ		1
τ_0		$1/\sqrt{2\sqrt{D}}$
τ_1		$1/\sqrt{2D}$
ρ		μ
R_1		GD
R_2		GI

©Springer-Verlag Berlin Heidelberg 2003

behavior of the evolution strategy in any case. This improvement is independent from other parameter settings of the underlying evolution strategy. Thus, there is no hint that the results were caused by interactions between other factors.

5 Summary and Outlook

The elevator supervisory group control problem was introduced in the first part of this paper. Evolution strategies were characterized as efficient optimization techniques: they can be applied to optimize the performance of an NN-based elevator supervisory group controller. The implementation of a threshold selection operator for evolution strategies and its application to a complex real-world optimization problem has been shown. Experimental design methods have been used to set up the experiments and to perform the data analysis. The obtained results gave first hints that threshold selection might be able to improve the performance of evolution strategies if the fitness function values are stochastically disturbed. The TS operator was able to improve the average passenger handling capacities of an elevator supervisory group control problem.

Future research on the thresholding mechanism will investigate the following topics:

- Developing an improved sampling mechanism to reduce the number of additional fitness function evaluations.
- Introducing a self-adaptation mechanism for τ during the search process.
- Combining TS with other statistical methods, e.g. Staggge's efficiently averaging method [Sta98].

Acknowledgments. T. Beielstein's research was supported by the DFG as a part of the collaborative research center 'Computational Intelligence' (531).

R, a language for data analysis and graphics was used to compute the statistical analysis [IG96].

References

[AB01] Dirk V. Arnold and Hans-Georg Beyer. Investigation of the (μ, λ)-ES in the presence of noise. In J.-H. Kim, B.-T. Zhang, G. Fogel, and I. Kuscu, editors, *Proc. 2001 Congress on Evolutionary Computation (CEC'01), Seoul*, pages 332–339, Piscataway NJ, 2001. IEEE Press.

[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.

[Bey01] Hans-Georg 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.

[BM02] T. Beielstein and S. Markon. Threshold selection, hypothesis tests, and DOE methods. 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 777–782. IEEE Press, 2002.

[BPM03] T. Beielstein, M. Preuss, and S. Markon. A parallel approach to elevator optimization based on soft computing. Technical Report 147/03, Universität Dortmund, 2003.

[BS02] Hans-Georg Beyer and Hans-Paul Schwefel. Evolution strategies – A comprehensive introduction. *Natural Computing*, 1:3–52, 2002.

[IG96] R. Ihaka and R. Gentleman. R: A language for data analysis and graphics. *Journal of Computational and Graphical Statistics*, 5(3):299–314, 1996.

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

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

[MAB$^+$01] Sandor Markon, Dirk V. Arnold, Thomas Bäck, Thomas Beielstein, and Hans-Georg 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] Sandor Markon. *Studies on Applications of Neural Networks in the Elevator System*. PhD thesis, Kyoto University, 1995.

[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.

[Rud98] Günter Rudolph. On risky methods for local selection under noise. In A. E. Eiben, Th. Bäck, M. Schoenauer, and H.-P. Schwefel, editors, *Parallel Problem Solving from Nature – PPSN V, Fifth Int'l Conf., Amsterdam, The Netherlands, September 27–30, 1998, Proc.*, volume 1498 of *Lecture Notes in Computer Science*, pages 169–177. Springer, Berlin, 1998.

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

[Sii97] M. L. Siikonen. *Planning and Control Models for Elevators in High-Rise Buildings*. PhD thesis, Helsinki Unverstity of Technology, Systems Analysis Laboratory, October 1997.

[Sta98] Peter Stagge. Averaging efficiently in the presence of noise. In A.Eiben, editor, *Parallel Problem Solving from Nature, PPSN V*, pages 188–197, Berlin, 1998. Springer-Verlag.

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

©Springer-Verlag Berlin Heidelberg 2003