

An Experimental Analysis of Evolution Strategies and Particle Swarm Optimisers using Design of Experiments

Oliver Kramer
University of Dortmund
FB Informatik, LS XI
44221 Dortmund, Germany
oliver.kramer@udo.edu

Bartek Gloger
University of Paderborn
Fakultät EIM-I, FG WBS
33098 Paderborn, Germany
gloger@upb.de

Andreas Goebels
University of Paderborn
Fakultät EIM-I, FG WBS
33098 Paderborn, Germany
labeo@upb.de

ABSTRACT

The success of evolutionary algorithms (EAs) depends crucially on finding suitable parameter settings. Doing this by hand is a very time consuming job without the guarantee to finally find satisfactory parameters. Of course, there exist various kinds of parameter control techniques, but not for parameter tuning. The Design of Experiment (DoE) paradigm offers a way of retrieving optimal parameter settings. It is still a tedious task, but it is known to be a robust and well tested suite, which can be beneficial for giving reason to parameter choices besides human experience. In this paper we analyse evolution strategies (ES) and particle swarm optimisation (PSO) with and without optimal parameters gathered with DoE. Reasonable improvements have been observed for the two ES variants.

Categories and Subject Descriptors

I.2.8 [Computing Methodologies]: Artificial Intelligence—*Problem Solving, Control Methods, and Search*

General Terms

Algorithms, Experimentation

Keywords

Evolution Strategies, Particle Swarm Optimization, Design of Experiments

1. INTRODUCTION

The algorithms which were developed on the basis of evolution are called Evolutionary Algorithms. They are randomised search heuristics, which are heavily used in the field of optimisation in addition to the classic optimisation approaches, which cannot be applied when a couple of search space features can not be guaranteed. Another type of algorithms besides EAs, which we will discuss, too, is the PSO algorithm. It also borrows principles from nature in order to perform its task.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

GECCO '07, July 7–11, 2007, London, England, United Kingdom.
Copyright 2007 ACM 978-1-59593-697-4/07/0007 ...\$5.00.

We will use a technique introduced from the field of engineering in order to tune the algorithms' parameters. The concept of experimental analysis and the Design of Experiments (DoE) is not new, not even in the field of computer science, but the statistical tools are not common to many EA users¹. We will use these tools for our experimental analysis.

This article is structured as follows. The DoE is introduced and motivated in the following section 2. It lays the foundation to understand the procedure of executing a design which is then shown. Afterwards, in section 3, the implemented algorithms are presented and the optimal parameters are acquired, see section 4. The paper closes with the evaluation of the experimental results in section 5.

2. DESIGN OF EXPERIMENTS

2.1 Performance Measures

Before we can compare two algorithms, we have to specify a measure for their comparison. There are many different measures for the quality of an algorithm, i.e. the quality of the best solution, the percentage of runs terminated successfully, the number of iterations or time steps required to obtain the results, or the robustness of the algorithm. In the following we will list some typical measures to demonstrate that there is no standard rule for EAs:

- The mean best fitness can be defined as the average value of the best fitness values found at termination for one specific run configuration. We compare our experiments mainly based on this measure and do that by using fitness graphs.
- The hitting time of an ϵ environment is applicable to real-world optimisation problems, where the exact optimum can only be approximated with finite accuracy. Schwefel proposes the definition of an explicit border to determine successful runs or failures [5].
- Robustness of an algorithm can be defined in many ways, i.e. as a good performance over a wide range of instances of one test problem or even over a wide range of different test problems. We test the robustness of the examined algorithms in the experimental runs with standard parameter settings.
- To measure the algorithm speed, the average number of evaluations to find a solution can be used. The

¹EA users are also called *practioners* in [2].

maximum number of evaluations can be used for runs finding no solutions.

2.2 Design of Experiments (DoE)

DoE is a part of a method known as *process modelling and improvement*. We will concentrate on the methods useful for EAs and explored by Bartz-Beielstein in [1]. We follow the guidelines described in [2].

In an experiment, we deliberately change one or more process variables (or *factors*) in order to observe the effect the changes have on one or more *response* variables. The design of experiments is an efficient procedure for planning experiments in a way that the data obtained can be analysed to yield valid and objective conclusions. The main assumptions in order to apply DoE to evolutionary algorithms are the following. The factors of a DoE are the parameters for an EA. The response can be defined as the quality of the results of the EA (e.g. average fitness value at a given generation, or its convergence ratio). DoE begins with the determination of the objectives of an experiment and selecting the factors for the study. An experimental design is the laying out of a detailed experimental plan in advance of doing the experiment. Well chosen experimental designs maximise the amount of *information* that can be obtained for a given amount of experimental effort.

It is common to begin with a process model of the *black box* type, with several discrete or continuous input factors that can be controlled, and one or more measured output responses. The output responses are assumed to be continuous. Experimental data are used to derive an empirical (approximation) model linking the outputs and inputs. The most common empirical models fit to the experimental data. They can take a linear form:

$$Y = \beta_0 + \beta_1 \cdot X_1 + \beta_2 \cdot X_2 + \beta_{12} \cdot X_1 \cdot X_2 + \epsilon$$

Here, Y is the response for given levels of the *main effects* X_1 and X_2 . The $X_1 \cdot X_2$ term is included to account for a possible interaction effect between X_1 and X_2 .

There are many uses of DoE. We will apply the DoE in order to achieve three goals:

- Selection of the main parameters with *screening experiments*
- Maximising our response value (getting to the optimum) with *Response Surface Modelling* (RSM).
- Finally comparing the EAs with the gained optimal values (comparative experiments).

It follows that this process can be classified as *parameter tuning* since we are tuning exogenous parameters. What the term process points at, is the idea that not one big experiment will give the answer but that iterative actions can achieve and reveal the intricacy of the problem. The objective determines the design model.

Now that we stated the objectives we will start with the description of the models used.

2.3 Screening Design

The primary purpose of the experiment is to select or screen out the few important main effects from the many less important ones. A common screening experimental design is one with all input factors set at two levels each. These

levels are called *high* and *low* or $+1$ and -1 , respectively. A design with all possible high/low combinations of all the input factors is called a *full factorial design* in two levels. Since the number of parameters in the considered EAs is relatively high, this design will not be applied from the start. The ES exhibits eight parameters which would lead to 256 test runs, repetition runs not counted. The solution to this problem is to use only a fraction of the runs specified by the full factorial design. Which runs to choose and which to leave out is the focus of the *fractional factorial design*. But as a designer you can use it as a tool and rely on the required factors to be correctly computed. We have used a statistical tool in order to create the experimental settings. It produces the essential plots and organises the data in a convenient manner.

The fractional factorial design makes it possible to have less experiments at the cost of aliasing. Depending on the *resolution* of the design² some factor interactions can be *confounded*, which means that we cannot differentiate between them. But as screening analyses the main effects only and possible interactions are investigated later, this is not a hindrance. The design resolution gives information about how badly the design is confounded.

Our first experiments are based on resolution III designs, which means that no main effect is confounded with any other main effect, but main effects can be confounded with two-factor interactions. The notation for a $k = 9$ parameter resolution III design which we will use is: 2_{III}^{9-5} . The meaning of the resolution levels is as follows: In resolution III designs main effects are confounded with two-factor interactions. In resolution IV designs no main effects are longer aliased with two-factor interactions, but two-factor interactions are aliased with each other and in resolution V designs no main effect or two-factor interaction is aliased with any other main effect or two-factor interaction, but two-factor interactions are aliased with three-factor interactions.

The basic purpose of a fractional factorial design is to economically investigate cause-and-effect relationships of significance in a given experimental setting. This does not differ in essence from the purpose of any experimental design. However, because we are able to choose fractions of a full design, and hence be more economical, we also have to be aware that different factorial designs serve different purposes.

Broadly speaking, with designs of resolution three, and sometimes four, we seek to screen out the few important main effects from the many less important others. For this reason, these designs are often termed *main effects designs*, or, as already stated, screening designs. On the other hand, designs of resolution five and higher are used for focusing on more than just main effects in an experimental situation.

There are a number of functional purposes for which designs are used. For example, an experiment might be designed to determine how to make a product better or a process more robust against the influence of external and non-controllable influences. Experiments might be designed to troubleshoot a process, to determine bottlenecks, or to specify which components of a product are most in need of improvement. Experiments might also be designed to optimise yield, or to minimise defect levels, or to move a process away from an unstable operating zone. All these aims and

²Which is noted in big roman letters.

purposes can be achieved using fractional factorial designs and their appropriate design enhancements.

2.4 Response Surface Design

Earlier, we stated that we will use the response surface method (RSM) objective. Under some circumstances, a model involving only main effects and interactions may be appropriate to describe a response surface when an analysis of the results revealed no evidence of *pure quadratic* curvature in the response of interest (i.e., the response at the center approximately equals the average of the responses at the factorial runs).

A *central composite design* (CCD) contains an imbedded factorial or fractional factorial design with center points that is augmented with a group of *star points* that allow estimation of curvature. If the distance from the center of the design space to a factorial point is ± 1 unit for each factor, the distance from the center of the design space to a star point is $\pm\alpha$ with $|\alpha| > 1$.

There are many factors which have to be determined before an experiment can start. There should be tests in order to avoid *floor* and *ceiling effects*. E.g. a ceiling effect happens when the optimum is reached within the machine dependant precision before the algorithm ends its search, or the algorithms is *stuck* in a local optimum for the larger amount of the algorithm's progress and consequently distorting the response value. The effects can be avoided, e.g. by a comparison of the distributions of successful runs: How many runs were completed successfully after t_{max} iterations? Or by varying the starting points: How many runs were completed successfully after t_{max} iterations from different starting points? Or even changing the problem dimension: How many runs were completed successfully after t_{max} iterations for different problem dimensions?

3. TESTED ALGORITHMS

This section describes the implemented algorithms.

3.1 Evolution Strategy with self-adaptation

The first considered EA is a $(\mu \dagger \lambda)$ -ES. Following features of the ES were chosen:

- Random parent selection with uniform distribution.
- Global intermediate and discrete recombination for both the objective variables and the strategy variables are possible.
- Mutation with self-adaptation of the parameters.
- Plus or comma selection of the best individuals.

This leads to the following algorithm specific factors [2]:

1. Number of parent individuals μ .
2. The selection pressure v : For given μ and v values, λ is the product of both and rounded to the nearest whole number.
3. Initial mean step sizes (standard deviations of the mutations of the decision variables).
4. Number of standard deviations n_σ , with $1 \leq n_\sigma \leq N$ where N denotes the problem dimension.
5. Global mutation parameter τ . Mutation is based on the extended log-normal rule, that enables learning mutation ellipsoids [3]:

$$\sigma^{(t+1)} = \exp(\tau_0 \cdot \mathcal{N}_0(0, 1))(\sigma_1^{(t)} \cdot \exp(\tau \cdot \mathcal{N}_1(0, 1)), \dots,$$

$$\sigma_N^{(t)} \cdot \exp(\tau \cdot \mathcal{N}_N(0, 1))$$

where $\sigma^{(t)}$ denotes the step length vector at iteration t , and $\mathcal{N}(0, 1)$ is the realisation of a normally distributed random number with variance 1 and expectation 0.

6. Number of parents ρ participating in crossover to create one offspring.
7. Recombination operator r_x for object variables: either intermediate or discrete values are possible.
8. Recombination operator r_s for the mutation strength.
9. Selection mechanism, maximum life span κ . Plus-strategies $(\mu + \lambda)$, and comma-strategies (μ, λ) can be generalised 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 ∞ , we model the plus-strategy.

Tabel 1 summarises the parameters of the ES we want to optimise with DoE.

3.2 Evolution Strategy with correlated mutation (ESC)

In addition to the self-adaptive mutation strength the ES with correlated mutation has got a number of angles which allow to rotate the ellipsoid. It helps in the convergence to an optimum and is supposed to be better on constrained multi-modal functions. Since the ascent of the fitness function is often not along the principal axes, it is the role of the correlation parameter to adapt to that. So, the representation of an individual changes to

$$\langle x_1, x_2, \dots, x_N, \sigma_1, \sigma_2, \dots, \sigma_N, \alpha_1, \alpha_2, \dots, \alpha_k \rangle$$

where $k = N \cdot (N - 1)/2$. Each α_i is an angle which represents a rotation in one of each of the planes between two dimensions. As in the basic form of the ES the strategy parameters are mutated and afterwards the objective parameters. The mutation of the strategy parameters stays the same. The rotation angles are mutated with a Gaussian standard distribution:

$$\alpha^{(t+1)} = \alpha + \beta \cdot \mathcal{N}(0, 1)$$

Schwefel recommends the value of β to be set to 0.0873 which corresponds to 5° . The value could leave the range of $[-\pi, \pi]$ so it is modified to stay in that range. The mutation of the individual is then:

$$x^{(t+1)} = x^{(t)} + \mathcal{N}(0, \sigma^{(t+1)}, \alpha^{(t+1)})$$

where $\sigma^{(t+1)}$ is the vector of strategy parameters and $\alpha^{(t+1)}$ the vector of k rotation angles. To construct the vector N one have to multiply a vector created from the mutated parameters $\Delta_{uncorr} = (\sigma_1^{(t+1)}, \dots, \sigma_n^{(t+1)})^T$ with k rotation matrices $R_{p,q}(\alpha_{p,q})$ where $\alpha_{p,q}$ is the angle of the plane spanned by the dimensions p and q .

3.3 Particle swarm optimisation

Each particle has a desire to exploit its own knowledge and a desire to explore the knowledge of other particles. We assume the particle has the following properties reflecting these desires:

- x : the particle's position
- v_{id} : current velocity
- p_{id} : best own position so far

Symbol	Parameter	Default Value	Range	Type
μ	Number of parents	15	\mathbb{R}	cont.
$v = \lambda/\mu$	Selection pressure (ratio offspring to parent)	7	\mathbb{R}_+	cont.
$\sigma^{(0)}$	Initial standard deviation	3	\mathbb{R}_+	cont.
n_σ	Number of standard deviations	N	$\{1, N\}$	nomin.
τ	Learning parameter	1	\mathbb{R}_+	cont.
ρ	Number of parents for each offspring	μ	$\{1, \mu\}$	nomin.
r_x	Recombination operator for objective variables	discrete	$\{d, i\}$	nomin.
r_s	Recombination operator for strategy variables	intermediate	$\{d, i\}$	nomin.
κ	Maximum age	0	$\{1, \infty\}$	nomin.

Table 1: Parameters of the ES which we want to optimise with DoE.

- p_{gd} : best position of other particles so far
- i : the individuality of the particle
- s : the social factor of the particle

These attributes lead to the following equation:

$$x' = x + v_{id} \quad (1)$$

and

$$\Delta v_{id} = i \cdot \phi_1 \cdot (p_{id} - x) + s \cdot \phi_2 \cdot (p_{gd} - x) \quad (2)$$

which determines the particles behaviour. It determines the change to the particles velocity in each dimension. The first product of the summation determines its desire to tend to its own best position found until now. The second product determines its desire to follow the global best value. The value of p_{gd} is determined through *neighbourhoods*. The variables ϕ_1 and ϕ_2 produce random numbers. The PSO algorithm implemented is configured with following standard parameters:

1. The number of particles in the swarm P . This has the biggest impact on the algorithm's performance.
2. The individuality of each particle and the associated boundaries. They determine the exploitation of the fitness landscape by each particle.
3. The sociality of each particle which counteracts the individuality and describes the tendency to follow the group.
4. The communication topology. Here, the star (s), the global (g), and the circle (c) topology are implemented.

Again, we summarise the exogenous parameters for which we want to find optimal settings, see table 2.

4. DETERMINING THE OPTIMAL PARAMETERS

In this section we want to apply the theory of DoE. The setup is as follows. We will try to optimise the parameter settings for the evolution strategy algorithm (ES) for all functions described in appendix A. We hope to discover a mutual good setting. In order to compare the three implemented algorithms we will use the DoE for the PSO, too, but as it is a time-consuming task, we will confine ourselves to only the Sphere and the Schwefel function. As a side-effect we can verify the results from [1]. In order to be able to compare the algorithms on the other two functions, we will use standard parameter settings for the test runs there.

Some preliminary test were made to determine the number of generations that create meaningful results. We want to prevent ceiling effects to influence the response, hence we set the number of generations for the Sphere function to 50 and for the other functions to 100.

4.1 The standard ES on the Sphere function

As was shown in [1] the standard parameter settings are not optimal. We have verified that with slightly different results. We set up the 2_{III}^{9-5} fractional factorial design that is used for all test functions. The setup for the 16 experiments is shown in Table 3. The actual values for each experiment can be seen in Table 4. After the 2_{III}^{9-5} fractional factorial

Problem	Sphere
Algorithm	ES(10; 20, 50; 200)
Number of runs	5
Initial standard deviation	1;5
Number of standard deviations	1;12
Learning parameter τ	1;2
Number of parents for each offspring	2;10;20 (binary or multi)
Recombination type of objective variables	discrete/intermediate
Recombination type of mutation strengths	discrete/intermediate

Table 3: Example for our experimental setups: The 2_{III}^{9-5} fractional factorial design on the Sphere function.

Experiment	μ	v	σ_0	n_σ	τ	ρ	X_{Rec}	S_{Rec}	κ
1	20	5	1	1	2	L1	L2	L2	L1
2	20	10	1	12	1	L1	L1	L2	L1
3	10	5	5	1	2	L2	L2	L1	L1
4	20	5	5	12	1	L1	L2	L1	L1
5	20	10	5	1	2	L1	L1	L1	L1
6	10	5	1	12	1	L2	L2	L2	L1
7	10	10	1	1	2	L2	L1	L2	L1
8	10	10	5	12	1	L2	L1	L1	L1
9	10	10	1	12	2	L1	L2	L1	L2
10	20	5	5	1	1	L2	L1	L2	L2
11	20	10	5	12	2	L2	L2	L2	L2
12	10	5	5	12	2	L1	L1	L2	L2
13	20	5	1	12	2	L2	L1	L1	L2
14	10	5	1	1	1	L1	L1	L1	L2
15	20	10	1	1	1	L2	L2	L1	L2
16	10	10	5	1	1	L1	L2	L2	L2

Table 4: The actual values for 16 experiments of the 2_{III}^{9-5} fractional factorial design on the Sphere function. L1 and L2 correspond to the respective high and low values

design it was clear that the parameters r_s, n_σ, v , and the interaction between μ and r_s have influence on the response of the model (see Figure 1). For each test, we have executed five runs. The normal probability plot is used to answer whether the data is normally distributed or what the nature of the variation from normality is. Here we can see that especially r_s diverges significantly. Another indicator for r_s is the box plot (Figure 2) and the pareto plot (Figure 4). A

Symbol	Parameter	Default Value	Range	Type
P	Number of particles	25	\mathbb{R}_+	cont.
i	The initial individuality of the particle	1	\mathbb{R}_+	cont.
s	The initial social factor of the particle	1	\mathbb{R}_+	cont.
Top	Topology used for the communication between the particles	g	s, g, c	nomin.

Table 2: Parameters of the PSO which we want to optimise with DoE.

pareto plot is a bar chart that displays the classification of problems arranged in decreasing order. The column whose values are the cause of a problem is assigned as Y and is called the process variable. The pareto plot produces charts to display the relative frequency or severity of problems in a quality-related process or operation. In Figure 3 a visual-

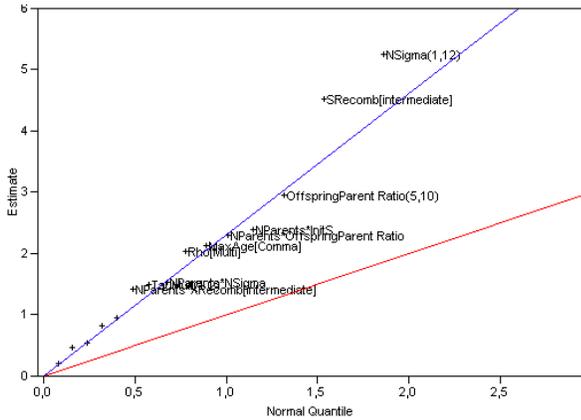


Figure 1: Half-normal plot for the Sphere function: The parameter r_s and the number of strategy parameters diverge from the half-normal.

isation of the response values in a cube plot are shown. The response values were used to create the following model. As only two parameters are significant on the 95% level, we decided to run a full factorial design with center points for the two parameters. In an experiment with quantitative factors, the experimental condition corresponding to all factors being set to the mid-point between their high and low values is called center point. Center points serve to test for the presence of curvature, and give information about quadratic effects. When repeated, center points also provide estimates of the magnitude of the experimental error.

4.2 The standard ES on the other functions

The optimal parameters for the other functions were determined in the same way as for the Sphere function: we started with a screening design, created plots to visualise the data and decided how to model the optimisation runs. The new model was analysed and the optimal parameters were determined.

The total number of runs per optimisation is: 16 for the screening design, 10 for the full factorial design to determine the best settings (or in the case of the Griewangk function 16, since there we decided to include one more parameter), and one final run for a comparison. The 26 DoE experiments had 5 repeats each whereas the final setting was run 50 times. So, altogether 180 runs per optimisation have been performed.

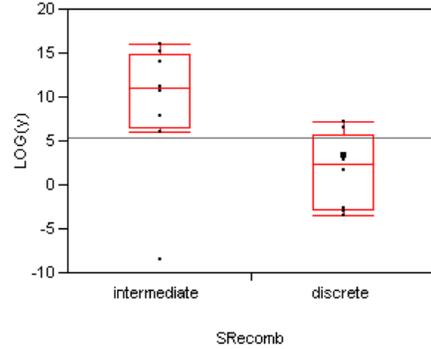


Figure 2: Box plot of the parameter r_s : The parameter r_s with the discrete setting leads to a lower response. Not even the outliers change the great difference in the means.

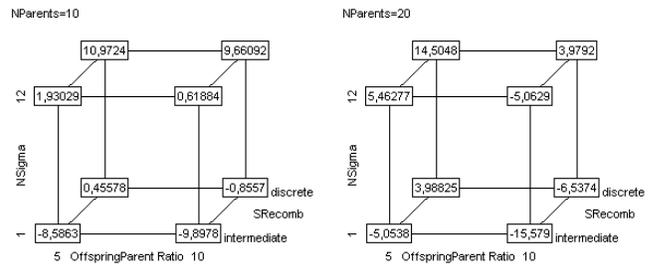


Figure 3: The four most significant parameters are shown and their combined response values are inscribed.

4.3 The PSO on all functions

For the optimisation of the PSO algorithm we decided to make a response surface model (RSM), as the number of parameters is low. We used the Box-Wilson³ design. This is an experimental design consisting of three parts:

- A two-level full or fractional factorial design
- *Star* or *axial* points in which each factor is varied to high and low levels with all other factors held constant
- center points

We performed the experiments on the Sphere function. We also conducted this design for the Schwefel function, but with no increase in the accuracy. In the following, the results of these experiments are presented.

³Star composite design

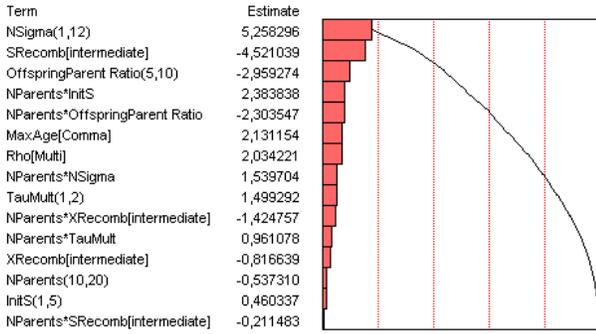


Figure 4: Pareto plot for the Sphere function: The parameter r_s (SRecomb) and the number of strategy parameters show the most influence on the model.

4.4 Optimal parameters

The optimal⁴ parameters are:

Sphere ES ($v = 10, n_\sigma = 1, \kappa = -1, \sigma_0 = 1, \tau = 1, X_{Rec} = intermediate, SRec = intermediate, \rho = binary, \mu = 20$)

Schwefel ES ($v = 20, n_\sigma = 30, \kappa = -1, \sigma_0 = 1, \tau = 2, X_{Rec} = discrete, SRec = discrete, \rho = multi, \mu = 20$)

Griewangk ES ($v = 10, n_\sigma = 1, \kappa = -1, \sigma_0 = 1, \tau = 1, X_{Rec} = intermediate, SRec = intermediate, \rho = binary, \mu = 20$)

Rastrigin ES ($v = 10, n_\sigma = 30, \kappa = -1, \sigma_0 = 1, \tau = 2, X_{Rec} = discrete, SRec = discrete, \rho = multi, \mu = 20$)

Sphere PSO ($N = 25, i = 1.55, s = 1.55, Top = global$)

Our hope to find a mutual good setting has not been fulfilled. The values of the optimal parameter settings are not satisfying since they indicate that the chosen region of interest was not broad enough so the extreme values are set to be the optimal values. Our version of the PSO algorithm does not lend itself to optimise well in its current form. The RSM model for the Schwefel function did not deliver any significant improvements in our region of interest. We assume that additional parameters like weighting the sociality factors depending on the current generation/step should make it possible to improve its performance. This shows that the DoE does not automatically mean that an improvement can be made. The result is highly dependent on the model and the available parameters.

5. EXPERIMENTAL RESULTS

In the following sections we will present some of the results we have obtained with the experiments. We will start with the outcomes from standard parameter settings and compare them on each function. Then we will present the results from the optimal parameter settings.

⁴The parameters we have determined are not that optimal as they could be because we have decided to stop examining the algorithms as we reached a significant improvement over the standard parameters. In that way they are only better.

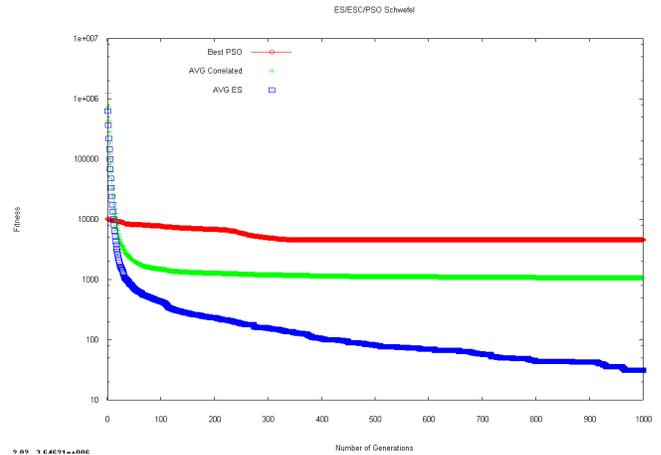


Figure 5: Experimental results with standard parameters of a PSO, a standard ES and an ES with correlated mutation on Schwefel's function. The PSO algorithm fails to find an optimum. Its best performance is shown.

5.1 Results with standard parameters

We have performed three experimental test series with standard parameters, see figure 5. Each algorithm was run on each problem function with standard parameter settings. The fitness graphs are always shown in logarithmic scale. On Schwefel's function the ES performs better on average than the ESC. The PSO is not able to find the optimum in any case.

On Griewangk's function again the ESC is on average worse than the ES. It is interesting to note that its best run is a magnitude better than the other algorithms' best runs. The PSO's inability to handle the accuracy is again visible but it is also interesting to note that until it reaches its maximum accuracy the algorithm delivers better results than the other two. The PSO beats both ES and ESC around generation 260, but is then *stuck*, as it reaches its granularity level.

The ESC performs on average worse on Rastrigin's function than the ES with standard self-adaptive mutation. The best performance of both, ES and ESC, in 50 runs showed that if the ESC somehow manages to adapt the correlated parameters, the convergence is much faster. It seems to be difficult for the ESC to adapt with the standard parameter settings. In the long run it has even a worse performance than the PSO.

While the problem of the PSO is that its minimum velocities are too high and because of that it cannot *slow down* on the optimum, the ESC is really *stuck*. The reason that the standard ES still converges on the minimum while the ESC does not, may be that both the rotation of the mutation ellipsoid is unfavourable and the number of parents is too low to adapt the angles sufficiently. The PSO algorithm has no self-adaptation, so it reaches the optima only within a threshold. Its velocities are too high and that causes the particles to pass the optima. A parameter scheme which is dependent on the actual generation could be applied in order to subside the velocities and therefore reach a more exact optimum.

To summarise these experiments, the ESC sometimes de-

livers excellent results, but performs on average worse than the ES. In the long run it is better than the PSO. What has to be noted in favour of the PSO is that the actual run time of the PSO is about ten times faster than the ES, while the ES is again perceptibly faster than the ESC.

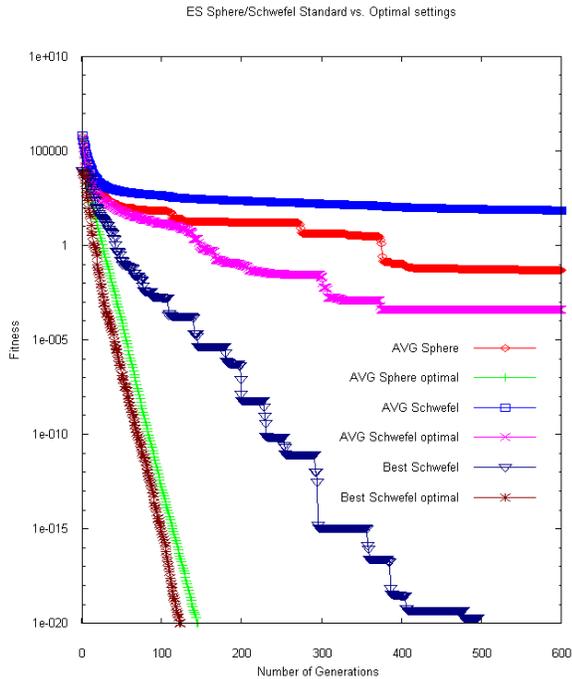


Figure 6: Optimisation with and without optimal settings. The experiments on the Sphere function worked out *as planned*, the average converges very fast. The convergence on the Schwefel function is on average better but still not optimal. Yet, the best run is a magnitude better than the best run with standard parameters.

5.2 Results with optimal parameters

We will examine the results of the experiments with optimal parameters in this section. Figure 6 shows the fitness graph of the optimal runs and the standard runs of the ES on the Sphere and Schwefel function. The optimisation process led to a very good convergence on the Sphere function. The algorithm converges on average to the optimum exponentially fast (a log scale is used). The best performance on the Schwefel function shows the same behaviour after the optimisation. There are no longer periods in which the algorithm resides in a local optimum as it is the case with the best not optimised run (Best Schwefel versus Best Schwefel optimal in Figure 6). Yet on average the algorithm still does not reveal the behaviour we hoped for. We have increased the success rate and lowered the ϵ environment of the optimum that the algorithm reaches on average.

A similar improvement can be observed on the Griewangk function (Figure 8). But this time the optimal parameter settings deliver a much better best run. The interdependence of the dimensions does not pose a problem to the ES. Figure 7 shows the improvement of the ES on the Rastrigin function. The worst run is still better than the average of fifty runs with standard parameters. Surprisingly the best

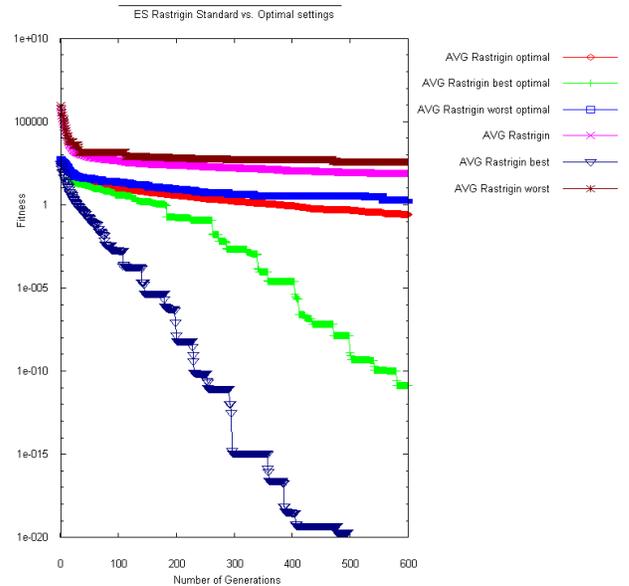


Figure 7: Although the best performance with standard parameters is still better than the best performance with optimal parameters, the worst run with optimal parameters is still better than the average run with standard parameters.

run performance with standard parameters is still better than the best performance with optimal parameters. This shows that there is still space for improvement. We do not consider the best runs as outliers because the environment is only pseudo random. The value would be produced again, if the experiment is repeated. We choose the random seed to be the same during each run.

The optimisation through the DoE has an impact on the performance of the PSO, though only of a constant factor (see Figure 9). The fitness curve of the optimised PSO seems to be shifted along the y-axis. Although its accuracy is improved and a better approximation to the optimum is found, the overall behaviour during the run has not changed. Whether it is an inherent attribute of the algorithm or our parameter search was not good enough stays an open question. We state that the PSO algorithm has been improved, but still does not arrive at the performance level of the not optimised performance of the ES.

6. CONCLUSIONS

The ES/ESC algorithms outperform the PSO in terms of convergence and accuracy. The reason for the poor performance of the PSO might be that we did not integrate additional control factors or further extensions. Another aspect that we did not inspect in this analysis is the runtime performance of the algorithms. Here, it is again noteworthy that the PSO excels. It finishes 50 experiments with 1000 steps within a minute. For the same amount of generations the ES needs about half an hour. Another interesting result is the curvature of the fitness graphs. The ES/ESC algorithms have usually a fast start and decelerate with the progress of generations. If an optimal setting is used, the solution will be found exponentially fast. The PSO algorithm reveals a reversed behaviour: its convergence speeds

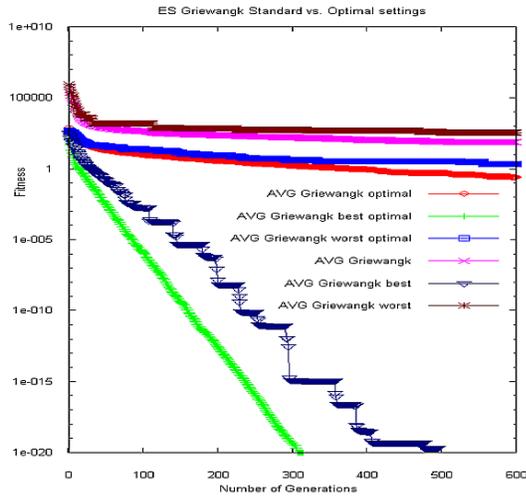


Figure 8: The ES on the Griewangk function. A logarithmically linear approximation of the optimum could only be achieved for the best runs with and without optimal settings.

up with the progress of the search. This could mean that it is not confined to a maximal convergence ratio like the ES appears to be. We suppose that the maximal convergence ratio originates in the mutation operator which modifies the objective variables with the *exp* function according to the log normal rule.

APPENDIX

A. TEST FUNCTIONS

The following test functions have been used.

Sphere Function.

$$f(x) = \sum_{i=1}^n x_i^2$$

The Sphere model is a continuous, strictly convex and unimodal function. The solution is at $x^* = (0, \dots, 0)$; $f(x^*) = 0$. This is a scalable problem in which we will use then dimension of $n = 12$ and the range of $|x_i| < 500$.

Schwefel's Function.

$$f(x) = 418.9829 \cdot n + \sum_{i=1}^n (-x_i \cdot \sin(\sqrt{|x_i|}))$$

The solution is at $x^* = (421, \dots, 421)$; $f(x^*) = 0$. Here we will use the dimension of $n = 30$ and the range of $|x_i| < 500$.

Rastrigin's Function.

$$f(x) = A \cdot n + \sum_{i=1}^n (x_i^2 - A \cdot \cos(2 \cdot \pi \cdot x_i))$$

This function has been used as a test function for distributed parallel ESs [4]. The solution is at $x^* = (0, \dots, 0)$; $f(x^*) = 0$ and we will use the range of $|x_i| < 5.12$ and the dimension of $n = 30$.

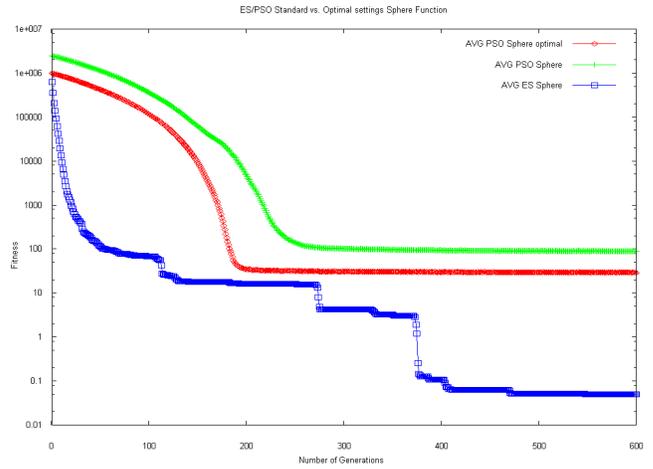


Figure 9: Even with the optimised parameters the PSO algorithm does not achieve the convergence of the average ES algorithm run.

Griewangk's Function.

$$f(x) = \frac{1}{4000} \cdot \sum_{i=1}^n (x_i^2) - \prod_{i=1}^n \left(\cos\left(\frac{x_i}{\sqrt{|i|}}\right) \right)$$

This function is taken from [6]. The solution is at $x^* = (0, \dots, 0)$; $f(x^*) = 0$. The range is chosen as $|x_i| < 500$ with $n = 30$ dimensions.

B. REFERENCES

- [1] T. Bartz-Beielstein. Experimental analysis of evolution strategies: Overview and comprehensive introduction. Technical Report Reihe CI 157/03, SFB 531, University of Dortmund, Dortmund, Germany, 2003.
- [2] T. Bartz-Beielstein. *New Experimentalism Applied to Evolutionary Computation*. PhD thesis, University of Dortmund, 2005.
- [3] H.-G. Beyer and H.-P. Schwefel. Evolution strategies – A comprehensive introduction. *Natural Computing*, 1:3–52, 2002.
- [4] G. Rudolph. Global optimization by means of distributed evolution strategies. In H.-P. Schwefel and R. Männer, editors, *Proceedings of the First Conference on Parallel Problem Solving from Nature (PPSN I)*, pages 209–213, Berlin, 1991. Springer.
- [5] H.-P. Schwefel. *Evolution and Optimum Seeking*. Sixth-Generation Computer Technology, Wiley Interscience, New York, 1995.
- [6] A. Törn and A. Pilinskas. Global optimization. In *Lecture Notes in Computer Science 350*. Springer-Verlag, 1989. Berlin Heidelberg.