

## On the efficiency of bionic optimisation procedures

S. Gekeler, R. Steinbuch & C. Widmann

*Reutlingen Research Institute, Reutlingen University, Germany*

### Abstract

Bionic optimisation strategies have proven to be efficient in many applications especially if there are many local maxima to be expected in parameter spaces of higher dimensions. In structural mechanics, the central question is whether one particular procedure is to be preferred generally or if there are different problem types where some procedures are more efficient than others. Evolutionary optimisation with some sub-strategies, particle swarm optimisation, and neural nets along with hybrid approaches that couple the aforementioned methods have been investigated to some extent. These approaches are not uniquely defined, but rather imply many variants regarding the definition and selection of next-generation members, varying parameters of the underlying processes and the criteria to switch the strategy. To measure the performance of the different approaches some simple test examples have been used. The indicator of the procedures performance was the number of individuals which needed to be studied in order to come up with a satisfactory solution. As our main concern was about problems with many optimisation parameters, artificial neural nets do not show sufficient convergence velocities in our class of optimisation studies. Evolutionary optimisation, its subclass of fern optimisation and particle swarm optimisation prove to be of comparable power when applied to the test problems. It should not be disregarded that for all these approaches some experience about the optimisation parameters has to be gathered. In consequence, the total number of runs or individuals necessary to do the final optimisation is essentially larger than the number of runs during this final optimisation. Good initial proposals prove to be the most important source for all optimisation processes.

*Keywords: bionic optimisation, evolutionary optimisation, particle swarm optimisation, performance, structural mechanics.*



# 1 Introduction

Optimisation deals with the modification of free parameters in given entities in order to come up with better values for given objectives or goals. In structural mechanics optimisation is often applied by the varying design data, e.g. the dimensions of a structure to improve the goal, e.g. to minimize the mass of a part or system or to reduce the energy consumption under service conditions.

The term “Bionic Optimisation” covers all the methods related to natural phenomena by which better variants of a given design are found. This implies that most natural processes use optimisation to adapt, survive and reproduce at given environmental conditions by better adapting to the situation. As there are infinite variants of applying this to technical problems, a strong classification of the observed phenomena is required in order to provide some understanding of the optimisation processes.

## 1.1 Terms and definitions

Before dealing with the different bionic optimisation strategies, some terms and definitions will help establish a common language. Care should be taken when reading papers by different authors or schools, as one term may be used for different subjects. Most authors accept that for an optimisation:

- We need a given *goal* or *objective*  $z$ .
- The objective  $z$  depends on a set of *free parameters*  $p_1, p_2, \dots p_n$ .
- There are defined *limits* and *constraints* on the parameters values.
- *Restrictions* of the parameter combinations or of derived data like stresses or displacements exclude unacceptable or unfeasible solutions.
- We want find the *maximum* (or *minimum*) of  $z(p_1, p_2, \dots p_n)$ .

Some discussion of the terms helps to better understand the following process.

- The *objective* or *goal* must be uniquely defined. We may not change the definition of the goal as this poses a new question and requires a new optimisation process.
- We need to find all parameters and their acceptable *value ranges* we might modify during the optimisation studies.
- The *parameter range* is the span of the free parameters' values given by lower and upper limits.
- The fewer free parameters we need to take into account, the faster the optimisation advances. Consequently, accepting some parameters as constant reduces the dimension of the solution space and accelerates the process.
- Restrictions like unacceptable system responses or unfeasible geometry must be taken into account. But occasionally restrictions limit the ranges of parameters to be searched. Such barriers have the potential to prevent the optimisation process from entering interesting regions.
- Finding the maximum of  $z(p_1, p_2, \dots p_n)$  is the same process as finding the minimum of the negative goal  $-z(p_1, p_2, \dots p_n)$ . In consequence, there is no need to distinguish between the search of maxima or minima.



## 1.2 Gradient or derivative based optimisation

Gradient or derivative based optimisation methods are the most popular ways to find improvements of given situations. From an initial position, the derivatives of the objective  $z(p_1, p_2, \dots, p_n)$  wrt. The free parameters are determined. Moving along this gradient has the tendency to yield local maxima in a small number of steps, as long as the search does not start too far away from the local maximum.

## 1.3 Bionic optimisation strategies

Bionic optimisation may be defined by many different approaches. Here we deal with some of the most commonly accepted classifications without taking into account all the many sub-classifications that might be found in the literature. The central approaches we are going to compare are

- Evolutionary Strategy (ES) [1, 2] - where paired or crossed parents have children by the combination and mutation of their properties. These children, or some of them, are parents in the next generation (Figure 1).

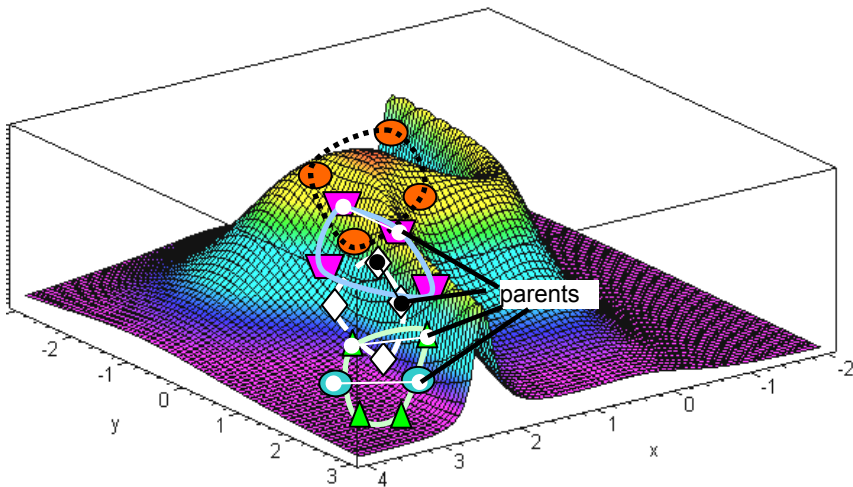


Figure 1: Evolutionary optimisation: 2 parents 4 kids.

- Fern Strategy (FS) - which may be regarded as a simplification of evolutionary optimisation. Individuals have offspring by mutation only but not by crossing properties with other members of the parent generation.

- Particle Swarm Optimisation (PSO) [3, 4] - where a population drifts through the possible solution space. The swarm's coherence is given by simple rules about the velocity of the individuals (Figure 2).

- Artificial Neural Nets (ANN) [6–8] - where training of the net yields an understanding of the solution space and allows the prediction of the system's response to given input. As ANN are not very efficient when applied to problems with many free parameters, we do not discuss them here [8].

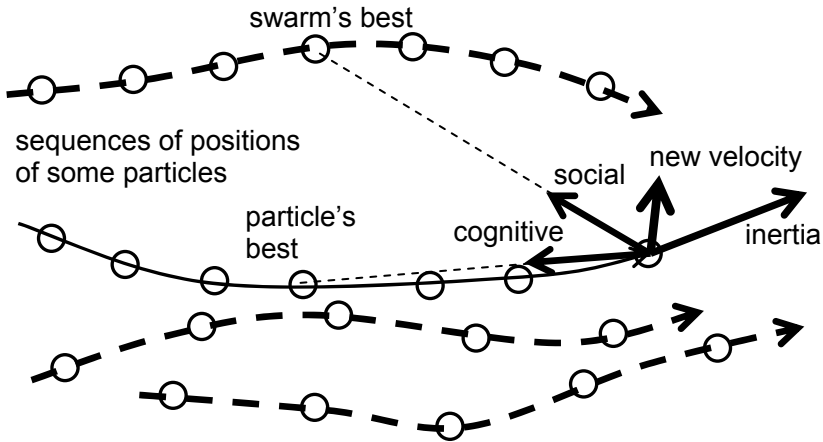


Figure 2: PSO, new velocity components of a particle.

#### 1.4 Measuring the efficiency of procedures

To quantify the efficiency of the different optimisation strategies, we have to introduce a measure that allows us to uniquely define the amount of work required to achieve a predefined quality. From some experience we propose to use the number of individuals to be analysed before coming close to an accepted good value. This requires the knowledge of what a good solution would be, which is generally not known as soon as we start studying new problems.

#### 1.5 Violation of boundary conditions

In all sequences of parameter sets that are based on random input, violations of the restrictions or boundary conditions may occur. Some problems are related to the fact that parameter combinations cause unfeasible geometries. Exceeding limits on physical responses, e.g. maximum stress or displacement have to be taken into account as well. There are different ways to deal with these unacceptable parameter sets.

The easiest way is to *remove all unacceptable individuals* from the list and to continue to produce members of the respective set until the required number of acceptable individuals is found. There is no reason not to use this selection type unless the cost of a specific function evaluation is too high to produce a number of individuals that may be essentially larger than the number of individuals required for the optimisation process.

Another way to keep the population near to the feasible range is to *punish all violation* of the given restrictions. A penalty multiplier weights the goal by the intensity of the violation. In consequence, punished individuals are less attractive for further reproduction, while the non-punished individuals have better chances to reproduce.

A third among many other ideas is to *fix the parameters* of the individual in violation to the border of the allowable space. This may be easily done for geometric input, but can be difficult if derived values like stresses or displacement are in question. In such cases, a reduction of the difference from a good individual's data may be used. If the parameters change less, the objective and the derived values may change less as well, so the violation may be avoided. We restrict our present study to the use of penalty functions for violations of the restrictions. The geometric input is set to the minimum or maximum value, if the randomly produced data exceed the respective limits. For PSO, we invert the particle's old velocity, if it violates given limits in addition to the penalty value. This combined approach has the advantage of simple applicability.

### 1.6 Hybrid strategies

In order to accelerate the optimisation process, it is not uncommon to use different strategies and to switch between them. This may be very efficient in many cases. We could start by using evolutionary optimisation so as to cover a region of the solution space and then change to particle swarms as soon as we feel we have reached the region of the best proposals [4]. During this study, where the main concern is about measuring the speed and efficiency of the different strategies, we avoid these hybrid methods, as their inclusion would lead to a large and confusing set of combinations and corresponding accelerations of the optimisation process.

## 2 Application of bionic optimisation strategies

The description of the approaches in section 1.2 needs to be improved upon to impart a qualified understanding of the studies.

- *Individuals* are the different elements of the parents and children sets.
- *Generation* is one step in the evolutionary processes, given by a set of parents.
- *Mutation* is the modification of an individual's parameters [2].
- The *mutation radius* is the maximum change of a parameter's value in a mutation step.

There are many other terms used in conjunction with bionic optimisation. As there is no generally accepted vocabulary, users are advised to check meanings carefully when reading papers from different authors [1, 2].

### 2.1 Evolutionary Strategy (ES)

In Evolutionary Optimisation two parents have one offspring by combining the properties of their DNA, here the values of the free parameters. This crossed DNA is subject to some random modification, the mutation. Some of the children will be better suited to adapt to the environmental challenges. Their chance to survive is superior to their siblings, so their genetic code becomes



dominant within the population. Therefore, these better children will be the next generation's parents. Some important things to note are:

- The *number of parents* should be sufficiently large to cover some or many possible parameter combinations, e.g. 0.5–5 times the number of parameters.
- The *number of children* should be 2–5 times the number of parents.
- *Pairing* selects two individuals to produce one common child.
- Should the *parents survive* to be parents in the next generation as well or not? Both approaches have their advantages and disadvantages.
- *Crossing*, the way by which two parents define the properties of one common child, and *mutation* may happen in different ways [2].
- The mutation radius may change during the course of the study
- *Selection* determines which Children of the current population (including the parents or not) should be the parents of the next generation.

ES tends to converge to the best solution if there are sufficiently large numbers of parents, children and generations and if the mutation radius is rather large. The number of individuals to be studied may become very large if the values driving the process are not set in a favourable range. Figure 3 plots the history of the goal of the three best and the worst parent for example F3 (cf. section 3.1).

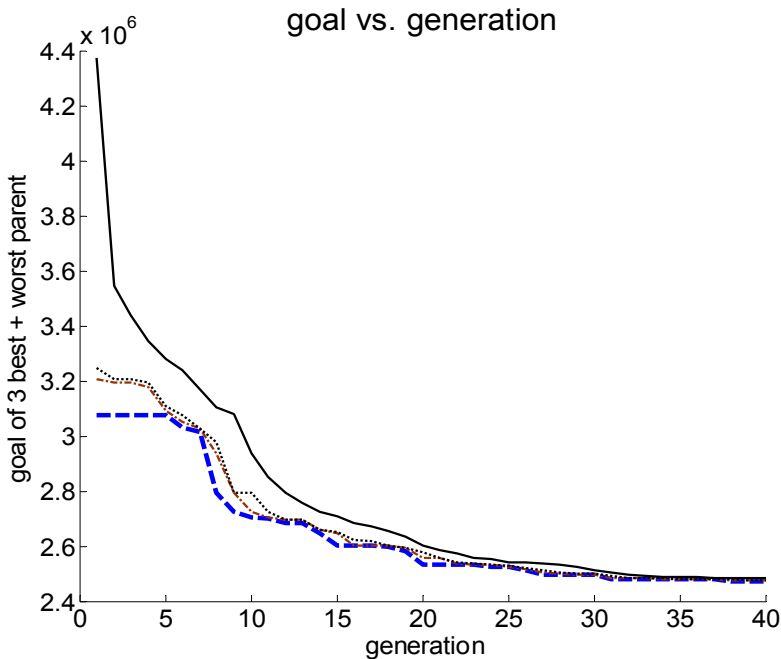


Figure 3: History of an evolutionary optimisation.

## 2.2 Ferns Strategy (FS)

Fern Optimisation is derived from the fact that ferns reproduce from the spores of one single parent. These spores do not have the same genetic properties as the parent. A certain mutation happens as in all duplication of the DNA.

- We define a limited number of initial designs - the parents - each given by a specific set of free parameters  $p_1, p_2, \dots, p_n$ .
- Each parent has a certain number of children, which are defined by a random mutation of their parent's parameters.
- This mutation radius may decrease during the course of the study.
- The best of the children will be the next generation's parents.
- This process is repeated for a certain number of generations.
- Sequences of parents and children which fail to come up with relatively good results are removed from the population to accelerate the process.
- The best individual of all offspring is the proposed optimum of the process.

As the solution space increases in size it becomes less probable that the coverage of the initial parts is sufficient to find very good designs. For smaller dimensions, FS is able to provide interesting results while remaining very easy to implement.

## 2.3 Particle Swarm Optimisation (PSO)

Particle Swarm Optimisation [3, 4] follows the observation that many groups of living beings have a tendency to behave like a complex being itself. The basic assumption of PSO is that the individuals know their position and velocity. In addition, they know where their best position during the process has been, and where the swarm's best position in the parameter space has been. Then the process is defined by the following (c.f. Figure 2):

- Each individual continues in its direction of travel: the inertia tendency.
- It tries to return to the best position it ever assumed: the cognitive tendency.
- It tries to approach the best position for all individuals: the social tendency.
- These tendencies are weighted by some specific ( $c$ ) and scalar or vector random ( $r$ ) values and added to the inertia vector [4, 5].

$$\mathbf{v}_{new} = c_v \mathbf{v}_{old} + c_{cog} r_{cog} \mathbf{d}_{cog} + c_{soc} r_{soc} \mathbf{d}_{soc} \quad (1)$$

- The particle's position in the next step is found by adding the new velocity to its old position. This corresponds to the definition of a new generation in the other approaches.

$$\mathbf{x}_{new} = \mathbf{x}_{old} + \mathbf{v}_{new} \quad (2)$$

PSO has proven to be very successful if an appropriate set of particles and velocity-weighting factors  $\{c_v, c_{cog}, c_{soc}\}$  has been used. Unfortunately, PSO has the tendency to stick to local minima if these parameters are not well chosen [5]. Figure 4 compares the results found at test example F3 for different weighting factors,  $c_v$ , and  $c_{soc}$  while the cognitive weight was  $c_{cog} = 0.1$ . There is a valley of efficient combinations of the coefficients. This valley is limited by steep hills indicating less efficient progress and weaker goals achieved by the PSO studies.

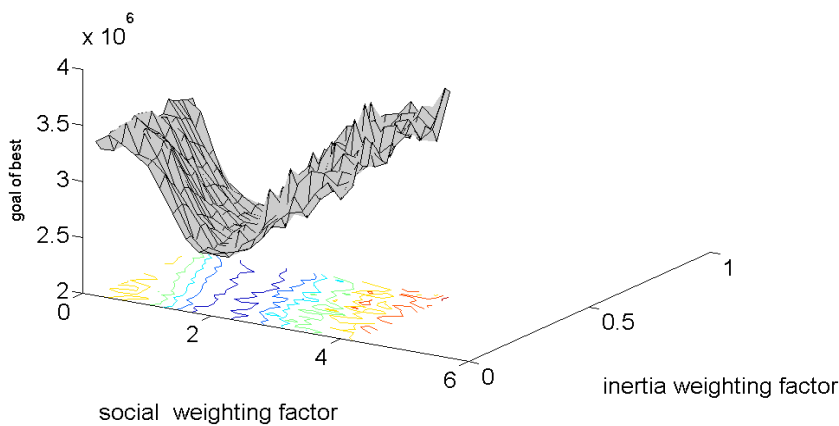


Figure 4: Dependency of the power of PSO of weighting factors for model F3.

### 3 Comparing the efficiency of bionic optimisation strategies

Optimisation is an expensive and time consuming process. We need to understand which procedure and which combinations of parameters may lead to a good and acceptable result within a reasonable amount of time.

#### 3.1 Test examples

Figure 5 depicts the 5 test examples used while Table 1 summarizes their data. We want to minimize the mass of the frames by varying the rods' cross sections without exceeding their maximum stresses and displacements. The grid size of the examples is 1000 mm except for example 2 where the grid size is 360 inches. Example F2 used imperial units (in, kip) the other frames use mm and Newton.

Table 1: Data of test problems.

frame	params	grid size	Amax / Amin	E-Mod	$\sigma_{\max}$	$d_{\max}$
F1	6	1000 mm	600 / 20 mm <sup>2</sup>	200 GPa	120 MPa	0.5 mm
F2	10	360 inch	35 / .1 inch <sup>2</sup>	10 Msi	25 ksi	2.0 inch
F3	13	1000 mm	400 / 20 mm <sup>2</sup>	200 GPa	50 MPa	0.5 mm
F4	58	1000 mm	400 / 20 mm <sup>2</sup>	200 GPa	100 MPa	2.0 mm
F5	193	1000 mm	600 / 20 mm <sup>2</sup>	200 GPa	450 MPa	20 mm

params: # of rods in frame

grid size: horizontal or vertical distance between the nodes

Amax, Amin: maximum and minimum allowed cross section area of the rods

E-Mod: Young's modulus

$\sigma_{\max}$ : maximum allowed stress in rod

$d_{\max}$ : maximum allowed displacement of nodes





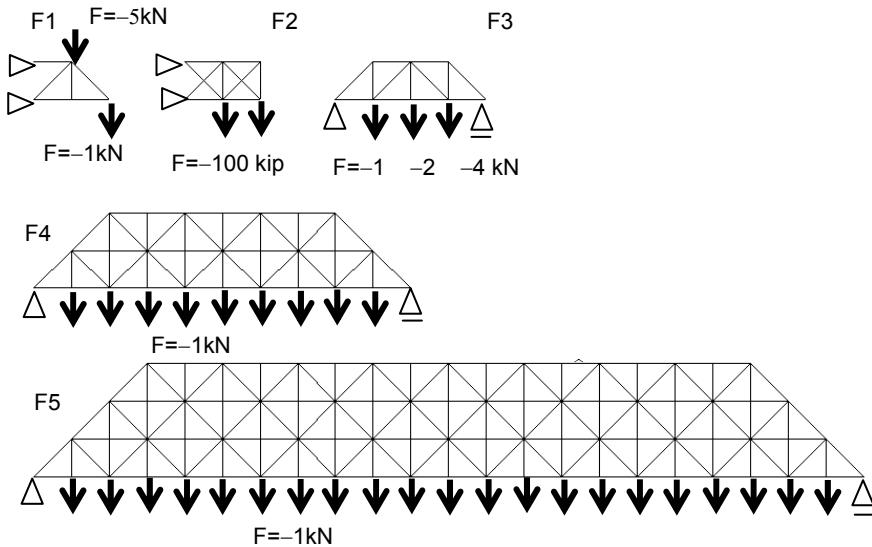


Figure 5: Test frames with loads and supports.

To come up with comparable results, we did a series of 20 loops for each problem and each strategy to avoid having only one or few very good or very bad results. On the other hand, the parameters we used were based on some experience with the underlying problems, so the number of runs presented does not come from naively starting a procedure, but includes some preliminary work which is impossible to quantify.

### 3.2 Input and results of the test examples

Table 2 lists the inputs of the test runs used. Table 3 and Figure 6 (individuals per loop) summarize the results of the test runs. The most important data are the number of individuals analysed to find a sufficient good design labelled as 'Individuals [1000]'. The number given multiplied by 1000 gives the total number of individuals required to find the proposed design. *mean* and *stddev* (standard deviation) and *best* are related to the results of the 20 runs. The ratio of the difference between the best and the average result divided by the standard-deviation (*reldev*) gives an idea of the stability of the strategy.

### 3.3 Interpretation of the results

ES, FS and PSO prove to be of a comparable efficiency when applied to the 4 smaller problems (F1, F2, F3, F4). Figure 6 indicates that there might be a nearly linear relation between the number of optimisation variables and the individuals required to find good proposals. For the largest problem F5 FS displays a performance that is essentially weaker than ES and PSO. ES and PSO seem to be

Table 2: Input parameters used.

## 2.1. ES

model	parents	kids	mut.rad.max	mut. Rad. min	generations
F1	10	20	.5	.05	60
F2	5	10	.5	.05	40
F3	5	10	.5	.05	50
F4	50	100	.5	.05	100
F5	100	200	.5	.05	200

## 2.2. FS

model	parents	kids/par.	mut.rad.max	mut. Rad. min	generations
F1	10	5	.5	.05	100
F2	10	4	.5	.05	50
F3	20	5	.5	.05	100
F4	100	5	.5	.05	200
F5	200	5	.5	.05	200

mutation radius reduced  
for ES and FS:

0% - 25% of generations:  $r_{mut} = .50$

25% - 50% of generations:  $r_{mut} = .20$

50% - 75% of generations:  $r_{mut} = .10$

75% - 100% of generations:  $r_{mut} = .05$

## 2.3. PSO

model	particles	generations
F1	10	30
F2	10	40
F3	20	80
F4	50	200
F5	800	70

Weighting factors:

$$c_v = 0.08, c_{cog} = 0.005, c_{soc} = 2.0$$

of comparable power when applied to the problem class which we discuss. FS shows promising results if the number of parameters is not too large, but the random search in high dimensional spaces becomes less successful. The scatter indicator *reldev* proposes that PSO has a more stable tendency to find solutions near the best while ES and FS show a larger range after the 20 runs.

Some knowledge may be gleaned from the results of these series of studies. Foremost that optimisation, especially bionic optimisation, is a process that consumes large amounts of time and computing power.

The results presented in section 3.2 would not have been found without a large number of preliminary studies providing experience in the field of optimisation of frames.

The input characteristics used in the test runs is derived from these preliminary studies. For example the selection of the 3 weighting factors  $\{c_v, c_g, c_s\}$  for the PSO required some 100 000 runs (Figure 4). The proposal of the reduction of the mutation range for ES and FS is the result of many studies as well. The proposal to use a number of initial parents in the size of free variables for ES and PSO is based on many studies, as well as the idea to use a large number of initial parents and a small number of children in FS.

Table 3: Results of 20 optimisation runs per problem.

strategy	model	mean	stddev	best	reldev	Individuals [1000]
ES	F1	1.62e6	.716e3	1.62e6	1.50	12
	F2	6.33e4	4.50e3	5.47e4	1.90	8
	F3	2.56e6	6.99e4	2.48e6	1.11	20
	F4	1.03e7	4.18e5	8.65e6	3.92	200
	F5	1.98e7	1.07e6	1.58e7	3.65	800
FS	F1	1.66e6	4.49e4	1.62e6	0.81	28
	F2	6.39e4	4.43e3	5.45e4	2.09	25
	F3	2.50e6	2.29e4	2.47e6	1.19	46
	F4	9.91e6	2.77e5	9.39e6	1.86	189
	F5	2.33e7	4.18e5	2.25e7	2.15	2570
PSO	F1	1.65e6	1.71e4	1.62e6	1.61	6
	F2	5.87e4	5.61e3	5.15e4	1.27	8
	F3	2.50e6	2.53e4	2.48e6	1.02	32
	F4	8.90e6	1.68e5	8.68e6	1.22	200
	F5	1.54e7	0.18e4	1.53e7	1.70	1120

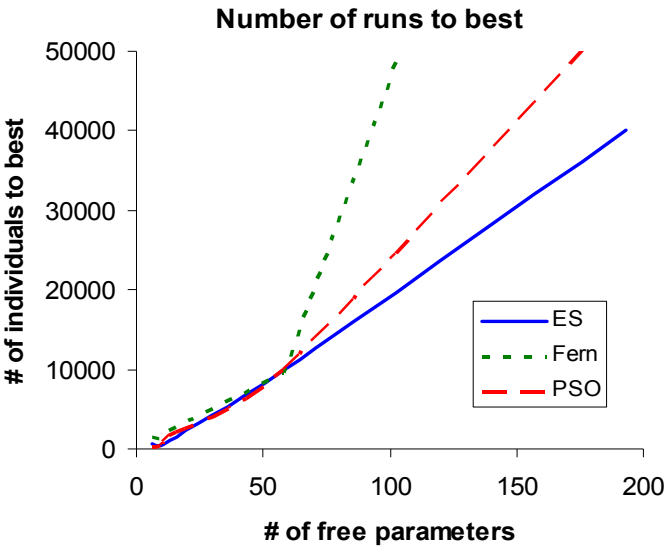


Figure 6: Efficiency of optimisation strategies.

One central fact about all optimisation may be learned from Figure 3. If there is a good initial design the number of optimisation runs to be done may decrease significantly. If an experienced engineer proposes an initial design with a goal of

e.g.  $2.6 \cdot 10^6$ , we need only 20 generations or 50% of the workload required to solve the task with a random initial design.

## 4 Conclusions

The quality of the initial proposals is the most important component of any optimisation. If experienced and motivated engineers propose designs that are close to the optimal ones there is a good chance that at least a local optimum will be found which is not too far away from the best solution possible. If we are close to good proposals, gradient methods will improve the parameters in short time and at reasonable effort.

As soon as we doubt that our initial designs are close to the optimal ones, ES or PSO have the capacity to propose of better designs. Nevertheless, the number of function evaluations may be large. Which of the two is to be preferred must be decided with some preliminary test. Often, the particle swarm shows a faster tendency towards the assumed best values, but some examples like those shown in Figure 4 indicate that the swarm might have the tendency to stick to local maxima like gradient methods.

Switching to gradient optimisation if a maximum is approached closely is always an interesting option. But experience has to be gathered there as well.

In every case, the optimisation of large problems is a time and resources consuming process. There is no way to avoid the evaluation of many individual solutions and there is no guarantee that the absolute best solution will be found at all.

## Acknowledgements

Part of this work has been supported by the federal government and some companies that we have been working with. We gratefully acknowledge their financial and engineering support.

## References

- [1] Rechenberg, I. *Evolutionsstrategie '94*, Frommann-Holzboog, Stuttgart, (1994).
- [2] Steinbuch, R. Successful Application of Evolutionary Algorithms in Engineering Design. *Special issue Vol 7, Journal of Bionic Engineering 7 Suppl* (2010), 199–210.
- [3] Coelho, L. D. S. and Mariani, V. C. Particle swarm optimization with quasi-Newton local search for solving economic dispatch problem, in *Proceedings—IEEE International Conference on Systems, Man and Cybernetics* (2007).
- [4] Plevris, V., Papadrakakis, M. A Hybrid Particle Swarm-Gradient Algorithm for Global Structural Optimization. *Computer-Aided Civil and Infrastructure Engineering* 26 (2011) 48–68.



- [5] Perez, R. E. and Behdinan, K. (2007b), Particle swarm optimization in structural design, in F. T. S. Chan and M. K. Tiwari (eds.), *Swarm Intelligence: Focus on Ant and Particle Swarm Optimization*, Itech Education and Publishing, Vienna, Austria, pp.373–94.
- [6] Berke L, Patnaik SN, Murthy PLN. Optimum design of aerospace structural components using neural networks. *Comput Struct* 48 (1993): 1001–1010.
- [7] Lagaros, N.D., Papadrakakis, M. Learning improvement of neural networks used in structural optimization. *Adv. in Engineering Software* 35 (2004) 9–25.
- [8] Widmann, Ch.: *Strukturoptimierung mit Neuronalen Netzen*, Master Thesis, Reutlingen University, (2012).

