Sequential Parameter Optimization (SPO) and the Role of Tuning in Experimental Analysis

Mike Preuss
Thomas Bartz-Beielstein

Algorithm Engineering
Universität Dortmund

EMAA Workshop
PPSN 2006, Reykjavik, September 9
Overview

1 Introduction
   Methodology Qualms
   Experimentation Elsewhere
   Better With Statistics?

2 Sequential Parameter Optimization
   Basics
   Overview
   Heuristic

3 Efficiency and Adaptability
   Parametrized Algorithms
   Beyond the NFL

4 This Is Not the End
   Just a First Step
Is Experimentation (in EC) Scientific?

Main goal of most investigations: Comparison of optimization algorithms

How do we generate performance data?
- 2 or more algorithms, *default* parameters
- Some test problems from a standard benchmark set
- Standard performance criterion

How do we compare?
- Traditional: Compare mean values
- Since about the 90s: significance tests (e.g. t-Test)

This gets us
- a) Some funny figures
- b) Lots of better and better algorithms which soon disappear again
Is Experimentation (in EC) Scientific?

Main goal of most investigations: Comparison of optimization algorithms
How do we generate performance data?
• 2 or more algorithms, *default* parameters
• Some test problems from a standard benchmark set
• Standard performance criterion
How do we compare?
• Traditional: Compare mean values
• Since about the 90s: significance tests (e.g. t-Test)

This procedure appears to be
a) Arbitrary (parameter, problem, performance criterion choice?)
b) Useless, as nothing is explained and generalizability is unclear

⇒ Do away with experimentation?
But, in many cases, theory building also fails
Goals in Evolutionary Computation

(RG-1) *Investigation*. Specifying optimization problems, analyzing algorithms. Important parameters; what should be optimized?

(RG-2) *Comparison*. Comparing the performance of heuristics


(RG-4) *Quality*. Robustness (includes insensitivity to exogenous factors, minimization of the variability) [Mon01]
Are We Alone (With This Problem)?

In natural sciences, experimentation is not in question

- Many inventions (batteries, x-rays, . . .) made by experimentation, sometimes unintentional
- Experimentation leads to theory, theory has to be *useful* (can we do predictions?)
- Theory idealizes (abstraction from the real world)

In computer science, the situation seems different

- 2 widespread stereotypes influence our view of computer experiments:
  a) Programs do (exactly) what algorithms specify
  b) Computers (programs) are deterministic, so why statistics?
Lessons From Other Sciences

In economics, experimentation was established quite recently (compared to its age)

- Modeling human behavior as the rationality assumption (of former theories) had failed
- No accepted new model available: Experimentation came in as substitute

In (evolutionary) biology, experimentation and theory building both have problems

- Active experimentation only possible in special cases (*drosophila et al.*)
- Otherwise only observation (passive experimentation)
- Mainly concepts (rough working principles) instead of theories: there are always exceptions
  ⇒ Stochastical distributions, population thinking

Nonlinear behavior

Ernst Mayr
Current “State of the Art” in EC

Around 40 years of empirical tradition in EC, but:

- No standard scheme for reporting experiments
- Still many *horse racing* papers
- Expressiveness (task?) and reproducibility often problematic
- Experimental methodology is just forming, including new statistical tools

Other sciences have more structured ways to report experiments, although usually not presented in full in papers. Why?

- Natural sciences: Long tradition, setup often relatively fast, experiment itself takes time (⇒ results valuable)
- Computer science: Short tradition, setup (implementation) takes time, experiment itself relatively fast (⇒ results volatile)
Statistical Methods and Their Pitfalls

- We claim: Fundamental ideas from statistics are misunderstood!
- For example: What is the $p$ value?

Definition ($p$ value)
The $p$ value is the probability that the null hypothesis is true.
Statistical Methods and Their Pitfalls

- We claim: Fundamental ideas from statistics are misunderstood!
- For example: What is the $p$ value?

**Definition ($p$ value)**
The $p$ value is the probability that the null hypothesis is true. **No!**
We claim: Fundamental ideas from statistics are misunderstood!

For example: What is the \( p \) value?

**Definition (\( p \) value)**

The \( p \) value is \( p = \Pr \{ \text{result from test statistic, or greater \mid null model is true} \} \)

\( \Rightarrow \) The \( p \) value is not related to any probability whether the null hypothesis is true or false.
New Concepts From the New Experimentalists

- Consider scientific meaning: Largest scientifically unimportant values
- Severe testing as a basic concept
- Observed significance level (OSL) plots to support testing
- First (*highly interdisciplinary*) Symposium on Philosophy, History, and Methodology of Error, June 2006
Components of an Experiment in EC

- Algorithm (program)
- Parameter set
- Test problem
- Performance measure
- Termination criterion
- Initialization
- Problem design
  - Test problem
  - Performance measure
  - Termination criterion
  - Initialization
Components of an Experiment in EC

- Algorithm (program)
- Parameter set
- Test problem
- Performance measure
- Problem design
  - Test problem
  - Performance measure
  - Termination criterion
  - Initialization

SPO mainly deals with:
- Algorithm design
- Algorithm (program)
- Parameter set

Induces:
- Control flow
- Data flow
Roots and Definitions

SPO integrates elements from Design of Experiments (DOE) and Design and Analysis of Computer Experiments (DACE) [SWN03]

- **Experiment** := optimization run
- **Design variables / factors** := parameters

- **Endogenous factors**: modified during the algorithm run
- **Exogenous factors**: kept constant during the algorithm run
  - Problem specific
  - Algorithm specific
SPO Overview

Phase I  Experiment construction
Phase II  SPO core: Parameter optimization
Phase III  Evaluation

- Phase I and III belong to the experimental methodology (how to perform experiments)
- Phase II is the parameter handling method, shall be chosen according to the overall research task (default method is provided)
- SPO is not *per se* a meta-algorithm: We are primarily interested in the resulting algorithm designs, not in the solutions to the primordial problem
SPO Workflow

1 Pre-experimental planning
2 Scientific thesis
3 Statistical hypothesis
4 Experimental design: Problem, constraints, start-/termination criteria, performance measure, algorithm parameters

5 Experiments
6 Statistical model and prediction (DACE). Evaluation and visualization
7 Solution good enough?
   Yes: Goto step 8
   No: Improve the design (optimization). Goto step 5

8 Acceptance/rejection of the statistical hypothesis
9 Objective interpretation of the results from the previous step
SPO Core: Default Method

Heuristic for Stochastically Disturbed Function Values

- Start with latin hypercube sampling (LHS) design: Maximum spread of starting points, small number of evaluations
- Sequential enhancement, guided by DACE model
- Expected improvement: Compromise between optimization (min $Y$) and model exactness (min $MSE$)
- Budget-concept: Best search points are re-evaluated
- Fairness: Evaluate new candidates as often as the best one

<table>
<thead>
<tr>
<th>$\lambda/\mu$</th>
<th>$\tau_0$</th>
<th>restart threshold</th>
<th>#eval best</th>
<th>config ID</th>
<th>result</th>
<th>std. deviation</th>
</tr>
</thead>
<tbody>
<tr>
<td>10.075</td>
<td>0.4180</td>
<td>22</td>
<td>4</td>
<td>42</td>
<td>0.0034</td>
<td>0.0058</td>
</tr>
<tr>
<td>5.675</td>
<td>0.7562</td>
<td>2</td>
<td>4</td>
<td>72</td>
<td>0.0042</td>
<td>0.0035</td>
</tr>
<tr>
<td>10.625</td>
<td>0.0796</td>
<td>5</td>
<td>4</td>
<td>57</td>
<td>0.0042</td>
<td>0.0054</td>
</tr>
<tr>
<td>4.905</td>
<td>0.1394</td>
<td>10</td>
<td>4</td>
<td>86</td>
<td>0.0047</td>
<td>0.0068</td>
</tr>
<tr>
<td>3.585</td>
<td>0.0398</td>
<td>13</td>
<td>4</td>
<td>81</td>
<td>0.0048</td>
<td>0.0056</td>
</tr>
<tr>
<td>3.145</td>
<td>0.0200</td>
<td>8</td>
<td>4</td>
<td>3</td>
<td>0.0050</td>
<td>0.0056</td>
</tr>
<tr>
<td>2.595</td>
<td>0.7960</td>
<td>4</td>
<td>4</td>
<td>83</td>
<td>0.0065</td>
<td>0.0048</td>
</tr>
<tr>
<td>2.375</td>
<td>1.8905</td>
<td>7</td>
<td>4</td>
<td>64</td>
<td>0.0113</td>
<td>0.0115</td>
</tr>
</tbody>
</table>

Table: Current best search points recorded by SPO, initial LHS
SPO Core: Default Method

Heuristic for Stochastically Disturbed Function Values

- Start with latin hypercube sampling (LHS) design: Maximum spread of starting points, small number of evaluations
- Sequential enhancement, guided by DACE model
- Expected improvement: Compromise between optimization ($\min Y$) and model exactness ($\min \text{MSE}$)
- Budget-concept: Best search points are re-evaluated
- Fairness: Evaluate new candidates as often as the best one

Table: Current best search points recorded by SPO, step 7

<table>
<thead>
<tr>
<th>$\frac{\lambda}{\mu}$</th>
<th>$\tau_0$</th>
<th>restart threshold</th>
<th>#eval best</th>
<th>config ID</th>
<th>result</th>
<th>std. deviation</th>
</tr>
</thead>
<tbody>
<tr>
<td>5.675</td>
<td>0.7562</td>
<td>2</td>
<td>4</td>
<td>72</td>
<td>0.0042</td>
<td>0.0035</td>
</tr>
<tr>
<td>10.625</td>
<td>0.0796</td>
<td>5</td>
<td>4</td>
<td>57</td>
<td>0.0042</td>
<td>0.0054</td>
</tr>
<tr>
<td>4.905</td>
<td>0.1394</td>
<td>10</td>
<td>4</td>
<td>86</td>
<td>0.0047</td>
<td>0.0068</td>
</tr>
<tr>
<td>3.585</td>
<td>0.0398</td>
<td>13</td>
<td>4</td>
<td>81</td>
<td>0.0048</td>
<td>0.0056</td>
</tr>
<tr>
<td>3.145</td>
<td>0.0200</td>
<td>8</td>
<td>4</td>
<td>3</td>
<td>0.0050</td>
<td>0.0056</td>
</tr>
<tr>
<td>2.595</td>
<td>0.7960</td>
<td>4</td>
<td>4</td>
<td>83</td>
<td>0.0065</td>
<td>0.0048</td>
</tr>
<tr>
<td>3.866</td>
<td>0.0564</td>
<td>4</td>
<td>8</td>
<td>106</td>
<td>0.0096</td>
<td>0.0065</td>
</tr>
<tr>
<td>2.375</td>
<td>1.8905</td>
<td>7</td>
<td>4</td>
<td>64</td>
<td>0.0113</td>
<td>0.0115</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>10.075</td>
<td>0.4180</td>
<td>22</td>
<td>8</td>
<td>42</td>
<td>0.0177</td>
<td>0.0181</td>
</tr>
</tbody>
</table>
SPO Core: Default Method

Heuristic for Stochastically Disturbed Function Values

- Start with latin hypercube sampling (LHS) design: Maximum spread of starting points, small number of evaluations
- Sequential enhancement, guided by DACE model
- Expected improvement: Compromise between optimization ($\min Y$) and model exactness ($\min \text{MSE}$)
- Budget-concept: Best search points are re-evaluated
- Fairness: Evaluate new candidates as often as the best one

<table>
<thead>
<tr>
<th>$\lambda$</th>
<th>$\mu$</th>
<th>$\tau_0$</th>
<th>restart threshold</th>
<th>#eval best</th>
<th>config ID</th>
<th>result</th>
<th>std. deviation</th>
</tr>
</thead>
<tbody>
<tr>
<td>10.625</td>
<td></td>
<td>0.0796</td>
<td>5</td>
<td>10</td>
<td>57</td>
<td>0.0024</td>
<td>0.0038</td>
</tr>
<tr>
<td>5.675</td>
<td></td>
<td>0.7562</td>
<td>2</td>
<td>5</td>
<td>72</td>
<td>0.0042</td>
<td>0.0031</td>
</tr>
<tr>
<td>4.905</td>
<td></td>
<td>0.1394</td>
<td>10</td>
<td>4</td>
<td>86</td>
<td>0.0047</td>
<td>0.0068</td>
</tr>
<tr>
<td>3.585</td>
<td></td>
<td>0.0398</td>
<td>13</td>
<td>4</td>
<td>81</td>
<td>0.0048</td>
<td>0.0056</td>
</tr>
<tr>
<td>3.145</td>
<td></td>
<td>0.0200</td>
<td>8</td>
<td>4</td>
<td>3</td>
<td>0.0050</td>
<td>0.0056</td>
</tr>
<tr>
<td>11.620</td>
<td></td>
<td>0.0205</td>
<td>2</td>
<td>10</td>
<td>111</td>
<td>0.0055</td>
<td>0.0052</td>
</tr>
<tr>
<td>2.595</td>
<td></td>
<td>0.7960</td>
<td>4</td>
<td>4</td>
<td>83</td>
<td>0.0065</td>
<td>0.0048</td>
</tr>
<tr>
<td>3.866</td>
<td></td>
<td>0.0564</td>
<td>4</td>
<td>8</td>
<td>106</td>
<td>0.0096</td>
<td>0.0065</td>
</tr>
</tbody>
</table>
SPO Core: Default Method

*Heuristic for Stochastically Disturbed Function Values*

- Start with latin hypercube sampling (LHS) design: Maximum spread of starting points, small number of evaluations
- Sequential enhancement, guided by DACE model
- Expected improvement: Compromise between optimization ($\min Y$) and model exactness ($\min \text{MSE}$)
- Budget-concept: Best search points are re-evaluated
- Fairness: Evaluate new candidates as often as the best one

<table>
<thead>
<tr>
<th>$\lambda$</th>
<th>$\mu$</th>
<th>$\tau_0$</th>
<th>restart threshold</th>
<th>$#\text{eval best}$</th>
<th>config ID</th>
<th>result</th>
<th>std. deviation</th>
</tr>
</thead>
<tbody>
<tr>
<td>10.625</td>
<td>0.0796</td>
<td></td>
<td>5</td>
<td>20</td>
<td>57</td>
<td>0.0023</td>
<td>0.0034</td>
</tr>
<tr>
<td>4.881</td>
<td>0.0118</td>
<td></td>
<td>8</td>
<td>20</td>
<td>116</td>
<td>0.0028</td>
<td>0.0029</td>
</tr>
<tr>
<td>5.675</td>
<td>0.7562</td>
<td></td>
<td>2</td>
<td>5</td>
<td>72</td>
<td>0.0042</td>
<td>0.0031</td>
</tr>
<tr>
<td>4.905</td>
<td>0.1394</td>
<td></td>
<td>10</td>
<td>4</td>
<td>86</td>
<td>0.0047</td>
<td>0.0068</td>
</tr>
<tr>
<td>3.585</td>
<td>0.0398</td>
<td></td>
<td>13</td>
<td>4</td>
<td>81</td>
<td>0.0048</td>
<td>0.0056</td>
</tr>
<tr>
<td>3.145</td>
<td>0.0200</td>
<td></td>
<td>8</td>
<td>3</td>
<td>3</td>
<td>0.0050</td>
<td>0.0056</td>
</tr>
<tr>
<td>11.620</td>
<td>0.0205</td>
<td></td>
<td>2</td>
<td>10</td>
<td>111</td>
<td>0.0055</td>
<td>0.0052</td>
</tr>
<tr>
<td>7.953</td>
<td>0.0213</td>
<td></td>
<td>2</td>
<td>10</td>
<td>114</td>
<td>0.0065</td>
<td>0.0055</td>
</tr>
</tbody>
</table>

Table: Current best search points recorded by SPO, step 17
SPO Core: Default Method

*Heuristic for Stochastically Disturbed Function Values*

- Start with latin hypercube sampling (LHS) design: Maximum spread of starting points, small number of evaluations
- Sequential enhancement, guided by DACE model
- Expected improvement: Compromise between optimization ($\min Y$) and model exactness ($\min \mathrm{MSE}$)
- Budget-concept: Best search points are re-evaluated
- Fairness: Evaluate new candidates as often as the best one

**Table:** Current best search points recorded by SPO, end (step 49)

<table>
<thead>
<tr>
<th>(\frac{\lambda}{\mu})</th>
<th>(\tau_0)</th>
<th>restart threshold</th>
<th>#eval best</th>
<th>config ID</th>
<th>result</th>
<th>std. deviation</th>
</tr>
</thead>
<tbody>
<tr>
<td>7.486</td>
<td>0.0329</td>
<td>13</td>
<td>50</td>
<td>140</td>
<td>0.0014</td>
<td>0.0022</td>
</tr>
<tr>
<td>6.367</td>
<td>0.0452</td>
<td>8</td>
<td>50</td>
<td>121</td>
<td>0.0015</td>
<td>0.0021</td>
</tr>
<tr>
<td>9.572</td>
<td>0.0536</td>
<td>11</td>
<td>50</td>
<td>134</td>
<td>0.0018</td>
<td>0.0031</td>
</tr>
<tr>
<td>6.024</td>
<td>0.0158</td>
<td>10</td>
<td>50</td>
<td>119</td>
<td>0.0019</td>
<td>0.0033</td>
</tr>
<tr>
<td>10.294</td>
<td>0.0229</td>
<td>8</td>
<td>50</td>
<td>133</td>
<td>0.0021</td>
<td>0.0036</td>
</tr>
<tr>
<td>6.798</td>
<td>0.0679</td>
<td>6</td>
<td>50</td>
<td>120</td>
<td>0.0021</td>
<td>0.0030</td>
</tr>
<tr>
<td>10.625</td>
<td>0.0796</td>
<td>5</td>
<td>50</td>
<td>57</td>
<td>0.0022</td>
<td>0.0032</td>
</tr>
<tr>
<td>4.8819</td>
<td>0.0118</td>
<td>8</td>
<td>20</td>
<td>116</td>
<td>0.0028</td>
<td>0.0029</td>
</tr>
</tbody>
</table>
SPO in Action

- Sequential Parameter Optimization Toolbox (SPOT)
- Introduced in [BB06]

Software can be downloaded from [http://ls11-www.cs.uni-dortmund.de/people/tom/ExperimentalResearchPrograms.html](http://ls11-www.cs.uni-dortmund.de/people/tom/ExperimentalResearchPrograms.html)
What is the Meaning of Parameters?

Are Parameters “Bad”?“

Cons:

• Multitude of parameters dismays potential users
• It is often not trivial to understand parameter-problem or parameter-parameter interactions
  ⇒ Parameters complicate evaluating algorithm performances

But:

• Parameters are simple handles to modify (adapt) algorithms
• Many of the most successful EAs have lots of parameters
• New theoretical approaches: Parametrized algorithms / parametrized complexity, (“two-dimensional” complexity theory)
Possible Alternatives?

Parameterless EAs:

- Easy to apply, but what about performance and robustness?
- Where did the parameters go?

Usually a mix of:

- Default values, sacrificing top performance for good robustness
- Heuristic rules, applicable to many but not all situations; probably not working well for completely new applications
- (Self-)Adaptation techniques, these cannot learn too many parameter values at once, and not necessarily reduce the number of parameters

⇒ We can reduce the number of parameters, but usually at the cost of either performance or robustness (or both)
⇒ We probably do not get rid of several parameters in most cases
Handling Parameters: Tuning and Comparison

*What do Tuning Methods (e.g. SPO) Deliver?*

- A spectrum of configurations, hinting at most important parameters and parameter interactions
- A best configuration of \( \{ \text{perf}(\text{alg}(\text{arg}^\text{ex}^t)) \mid 1 \leq t \leq T \} \) for \( T \) tested ones
- A progression of current best tuning results

![Graph showing LHS spectrum for spam filter problem](image)

![Graph showing current best configuration accuracy for spam filter](image)
Objections Against Parameter Tuning

... and How to Meet them (Hopefully)

a) The meta-algorithm (1. optimize parameters of an algorithm which is 2. used to tackle the original problem) is subject to the NFL\(^1\) (next slides)

b) Parameter optimization is too expensive

Possible solutions for b):
- Even a very small sample over the parameter space can help
- For recurring problems, parameter optimization eventually pays off
- Parameters may be optimized using simplified proxy problems (algorithm-based validation)

\(^1\) no free lunch theorem
The Art of Comparison

Orientation

The NFL told us things we already suspected:

- We cannot hope for the one-beats-all algorithm (solving the general nonlinear programming problem)
- Efficiency of an algorithm heavily depends on the problem(s) to solve and the exogenous conditions (termination etc.)

In consequence, this means:

- The posed question is of extreme importance for the relevance of obtained results
- The focus of comparisons has to change from:

  *Which algorithm is better?*

  to

  *What exactly is the algorithm good for?*
The Art of Comparison

Efficiency vs. Adaptability

Most existing experimental studies focus on the efficiency of optimization algorithms, but:

- Adaptability to a problem is not measured, although
- It is known as one of the key advantages of EAs

Interesting, previously neglected aspects:

- Interplay between adaptability and efficiency?
- How much effort does adaptation to a problem take for different algorithms? Or problems?
- What is the problem spectrum an algorithm performs well on?
- Systematic investigation may reveal inner logic of algorithm parts (operators, parameters, etc.)
Adaptability to a (One) Problem

Some Simple Measures

- \(\text{mean}(\text{LHS}(T)) \approx \text{expected performance with random parameter set}\)
- \(\text{best}(\text{LHS}(T)) \approx \text{expected performance for best of random search}(T)\)
- \(\text{best}(\text{SPO}(T_s)) \approx \text{performance of best existing parameter set}\)
Adaptability to a (One) Problem

Some Simple Measures

- \( \text{mean}(\text{LHS}(T)) \approx \text{expected performance with random parameter set} \)
- \( \text{best}(\text{LHS}(T)) \approx \text{expected performance for best of random search}(T) \)
- \( \text{best}(\text{SPO}(T_s)) \approx \text{performance of best existing parameter set} \)

Preuss/Bartz-Beielstein (Universität Dortmund)
Adaptability to a (One) Problem

*Some Simple Measures*

- $\text{mean}(\text{LHS}(T)) \approx \text{expected performance with random parameter set}$
- $\text{best}(\text{LHS}(T)) \approx \text{expected performance for best of random search}(T)$
- $\text{best}(\text{SPO}(T_s)) \approx \text{performance of best existing parameter set}$

![Bar charts showing spectrum: LHS, algorithm 1, 2, and 3](image)
Adaptability to a (One) Problem

Some Simple Measures

- \( \text{mean}(\text{LHS}(T)) \approx \text{expected performance with random parameter set} \)
- \( \text{best}(\text{LHS}(T)) \approx \text{expected performance for best of random search}(T) \)
- \( \text{best}(\text{SPO}(T_s)) \approx \text{performance of best existing parameter set} \)

Preuss/Bartz-Beielstein (Universität Dortmund)

SPO / Tuning / Experimental Analysis

PPSN 2006, Reykjavik, September 9
Empirical Findings

Concerning the example:

- The spectra are quite similar. Are the algorithms?
- Indeed. Only the mutation adaptation operators are different.

In general:

a) Some parameter sets do not work at all

b) An often found situation:
   - \( \frac{1}{3} \) of parameter sets lead to very bad performance
   - \( \frac{1}{3} \) are in the "interesting" performance region (good)
   - \( \frac{1}{3} \) are somewhere inbetween (not really interesting)

c) The performance potential SPO can reveal heavily depends on the algorithm, but with absolute distance parameters it works especially well

d) Sometimes adaptability appears to be exhausted after testing a relative small LHS design (⇒ low adaptability?)
Adapting EAs to Two Related Problems

100 peaks problem

spectrum: LHS, hillclimber EA

spectrum: LHS, swn-topology EA

spectrum: LHS, niching EA

spectrum: LHS, generic EA

reached performance (minimization) fraction in %

0.00 0.05 0.10

0 10 20 30
Adapting EAs to Two Related Problems

100 peaks problem

- Spectrum: SPO, hillclimber EA
- Spectrum: SPO, swn-topology EA
- Spectrum: SPO, niching EA
- Spectrum: SPO, generic EA

Reached performance (minimization) fraction in %

Preuss/Bartz-Beielstein (Universität Dortmund)
Adapting EAs to Two Related Problems

10 peaks + plateaus problem

spectrum: LHS, hillclimber EA

spectrum: LHS, swn-topology EA

spectrum: LHS, niching EA

spectrum: LHS, generic EA

fraction in %

reached performance (minimization)
Adapting EAs to Two Related Problems

10 peaks + plateaus problem

- spectrum: SPO, hillclimber EA
- spectrum: SPO, swn-topology EA
- spectrum: SPO, niching EA
- spectrum: SPO, generic EA

reached performance (minimization)
How do Tuning (SPO) Results Help?  
...or Hint to new Questions

What we get:

- A near optimal configuration, permitting top performance comparison or an estimation of "adaptability potential"
- A quality estimation of any previously (manually) found parameter set

No excuse: A first impression may be attained by simply doing an LHS

Yet unsolved problems:

- How much amount to put into tuning (fixed budget, until stagnation)?
- Where shall we be on the spectrum when we compare?
- Can we compare spectra (⇒ adaptability)?
- How to define adaptability as a measurable size?
