Automatic Parameter Tuning for Metaheuristics

Felix Dobslaw,
Information Technology and Media (ITM) Department
Östersund, Sweden

Mittuniversitetet
MID SWEDEN UNIVERSITY

Outline

- My home university and me
- Automatic Parameter Tuning
  - I. Overview
  - II. Default, Robust Parameter Settings
  - III. Proposed Methodology to Model-Based Optimization.

My University…

- Mid Sweden University (2005-)
  - Information Technology and Media Department (ca. 100 employees)
- International Masters
- Ca. 1.000 employees
- Ca. 16.000 students
- Multi-campus
- Distance education

Mittuniversitetet
MID SWEDEN UNIVERSITY

Mid Sweden University: Östersund, Sundsvall, Härnösand

Mittuniversitetet
MID SWEDEN UNIVERSITY

Östersund

Me

- 2005-2009 Software developer for IBM partner
  - Java, Plug-in, Portal, Database tooling
- March 2007 BSc in Informatics (University of Hamburg)
  - Broad Introduction
  - Thesis: Architecture for Information Retrieval Software
- April 2009 MSc in Computer Science (Mid Sweden University)
  - Specialized: Distributed Systems
  - Thesis: An Extendable Context Model for Distributed Sensor Networks
- June 2009-201? Phd studies (Mid Sweden University)
  - Focus on Automated Parameter Tuning

How I keep myself busy

- Research (80%):
  - Swarm Intelligence
  - Parameter Tuning
  - Learning:
    - Pedagogics(AI course)
    - Integer Programming
  - Context Modeling
- Teaching (20%):
  - AI Introduction
  - XML Introduction
  - AI advanced labs
  - Thesis supervision
- Sports:
  - Swimming
  - Jogging
  - Fitness
- Leisure:
  - Nature
  - Seeing friends
  - Board games
  - Traveling
  - Social Entrepeneurship…

Why am I here?

- Got in touch with Martin Holena
  - Surrogate Modeling via Neural Networks
- Getting to know
  - researchers, research students
  - another working environment
  - research approach(?)
- If possible, initiate cooperation.
- Talk here today!
- Present my work at the ”week of doctoral students 2010”.
- I will be here the whole next week. Looking forward to discussions!

Publications so far

A Parameter Tuning Framework for Metaheuristics Based on Design of Experiments and Artificial Neural Networks, Felix Dobslaw at ICCMN 2010: Rom, 27-30.04.2010

An Object-Oriented Model in Support of Context-Aware Mobile Applications, Felix Dobslaw, Aron Larsson, Theo Kanter, Jamie Walters at Mobilware Chicago, 29.06-02.07.2010

An Experimental Study on Robust Parameter Settings,

Felix Dobslaw at GECCO 2010 Graduate Student Workshop, Portland, 07.07.2010-11.07.2010

I.Parameter Tuning: Overview

I. Parameter Tuning
    I. Model-free vs. Model-based
    II. Sequential vs. Parallel
    III. Parameter Types
II. State of the art methods
    I. Sequential
    II. Parallel

I.I Parameter Tuning

Definition by Eiben et. al. (1999):

Parameter Tuning is the finding of good values for Parameters *before* the run of the algorithm.

I.I The Problem in Theory

- NP-hard combinatorial problems. (state-space explosion)
- Not trivial:
  - Near-optimal settings are problem (and of course algorithm) specific

AND

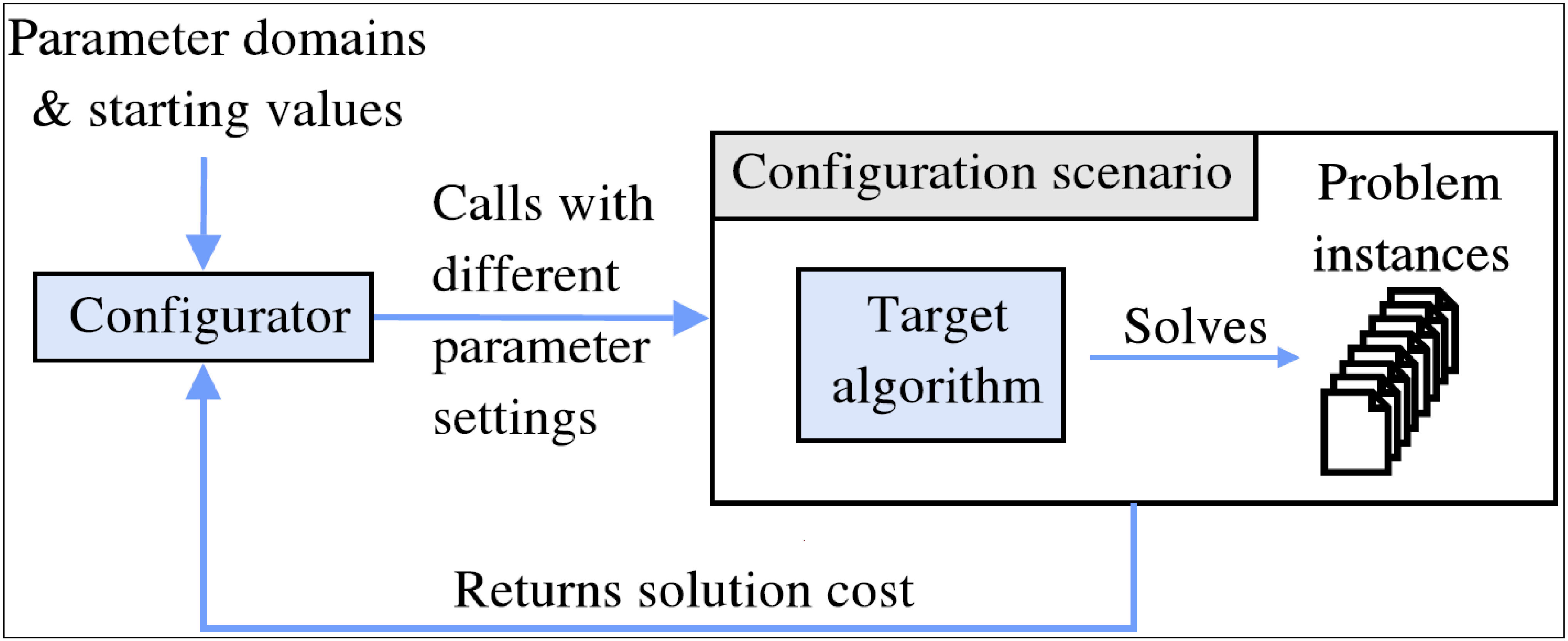
-> There is no free lunch!

I.I What kinds of Algorithms?

- Metaheuristics
  - Particle Swarm Optimization, Ant Colony Optimization, EA, Simulated Annealing Tabu Search, CPLEX

One objective: fairer comparison of algorithms!

I.I Parameter Tuning



From Hutter et. Al. (2009)

I.I.I Model-free vs. Model-based Parameter Tuning

Model-free:

- Simple algorithms without a ”memory”.

Model-based:

- Response Surface Model for
  - Deciding upon new regions to assess
  - Recommending settings for unseen problem instances

Mittuniversitetet
MID SWEDEN UNIVERSITY

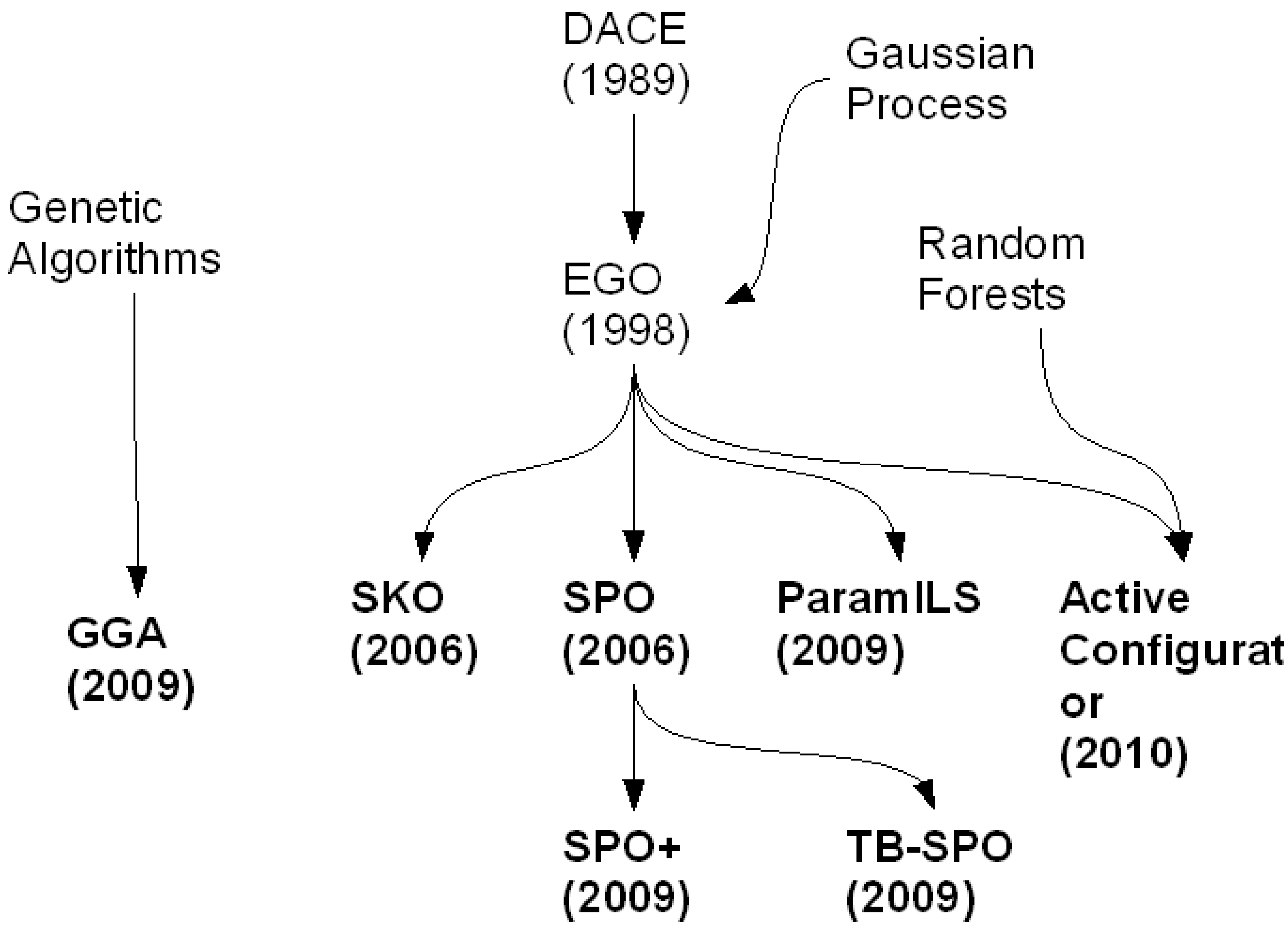
I.I.II Sequential vs. Parallel Model-Based Parameter Tuning

- Sequential tuning:
  - One refinement at a time
- Parallel tuning:
  - Multiple refinements at a time
  - E.g. Population based algorithms

I.I.III.Parameter Types

- Continuous (real numbers)
- Discrete (Integer Programming: e.g. TSP)
- Cagetorial (e.g. Selection mechanism in EA)

I.IV.State of the Art Methods (paper 3)

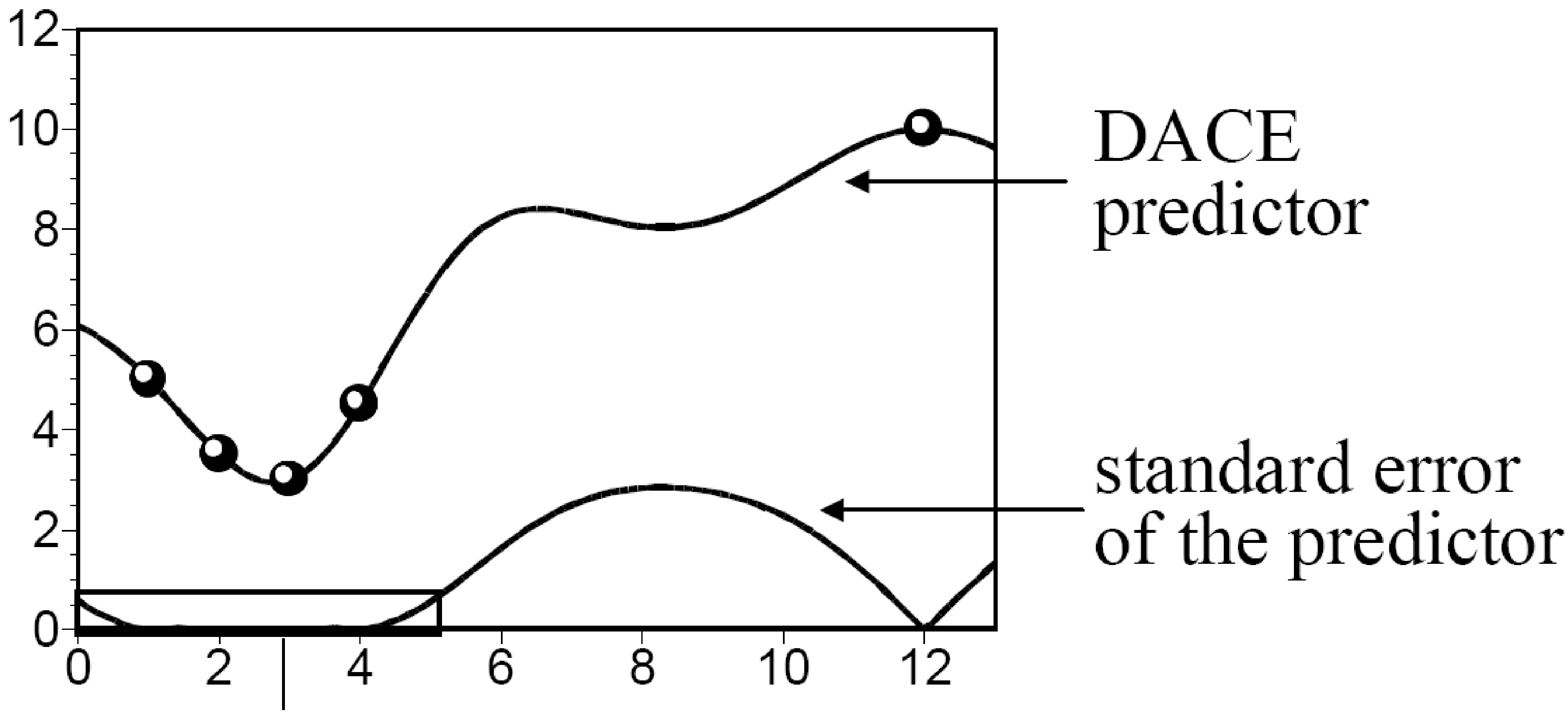


I.IV.Design and Analysis of Computer Experiments DACE

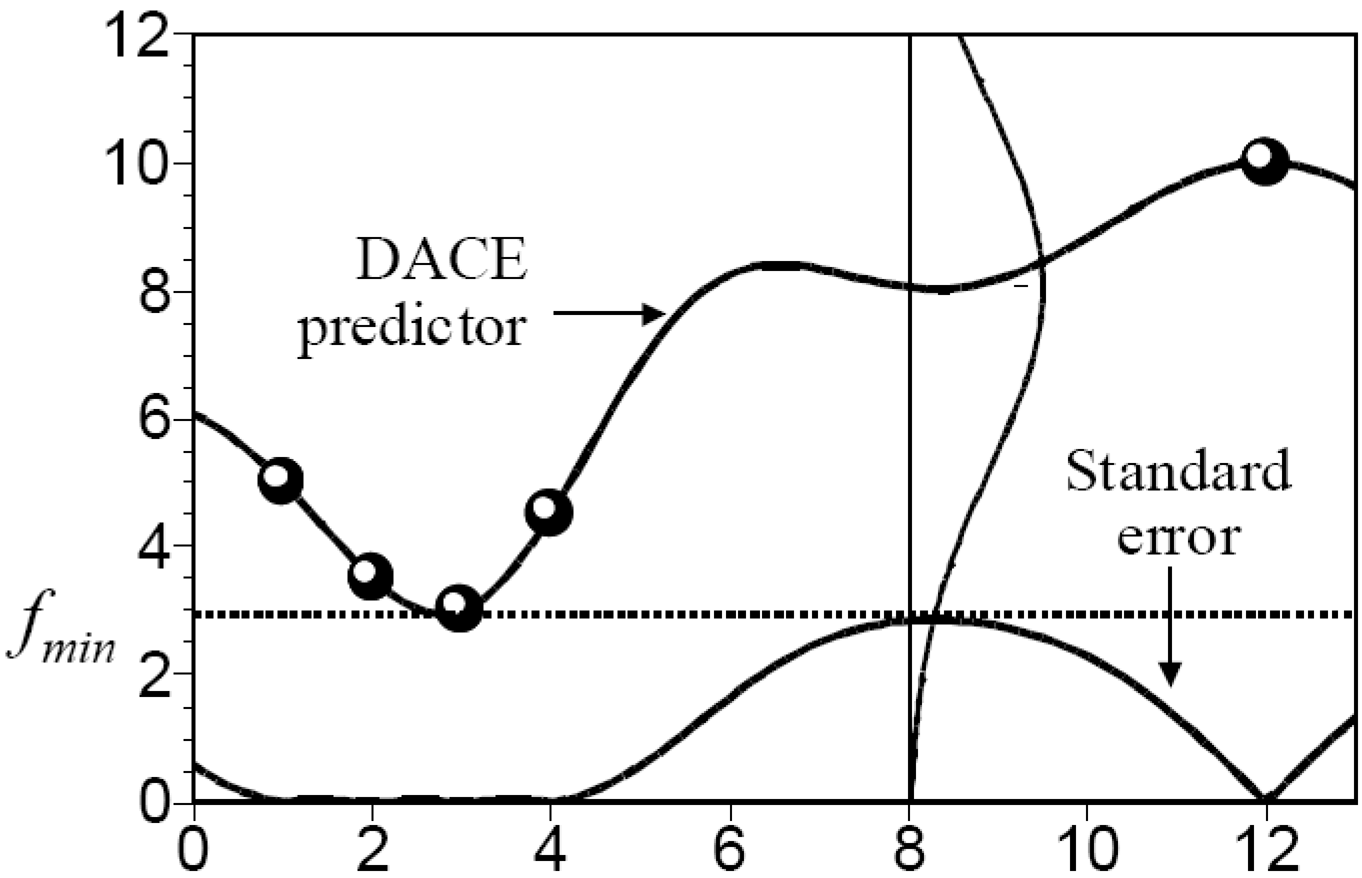
- Model on deterministic computer simulations
- Regression model with random process Z

$$Y(x) = \sum_{j=1}^{k} \beta_j f_j(x) + Z(x)$$

I.III Black Box optimization and EGO



I.III Black Box optimization and EGO



I.III Black Box optimization and EGO

1. Fit a DACE model (n=10k rule)
2. Maximize estimated improvement e (branch and bound) until e<1%.
   a) If not converts: re-estimate DACE model and go to 2.

I.III Black Box optimization and EGO

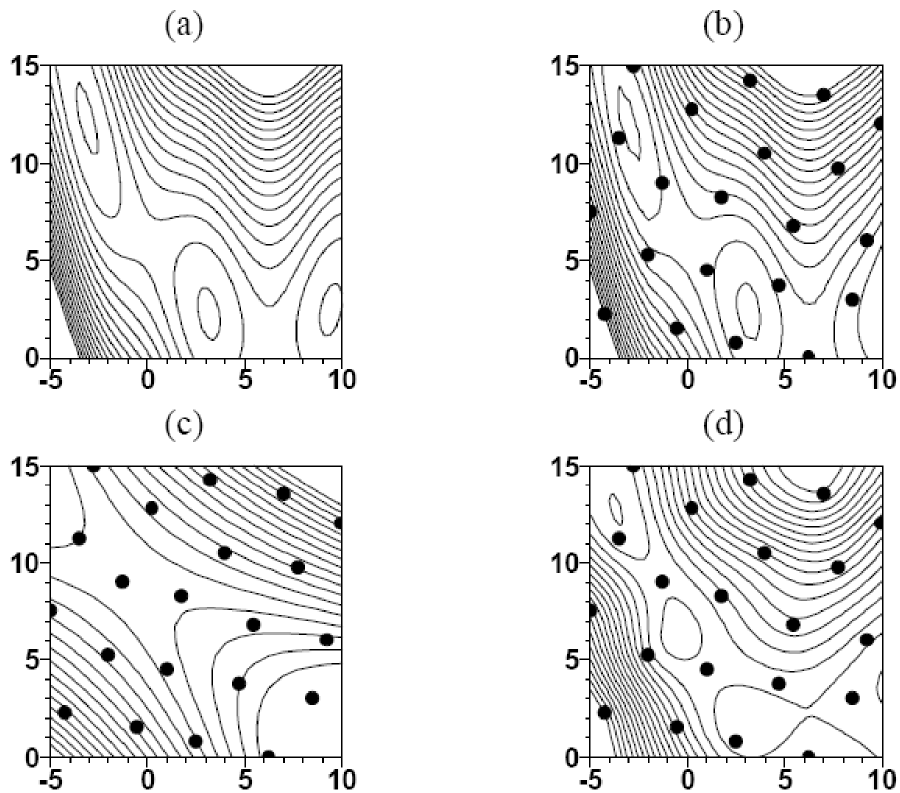


Figure 3. (a) Contours of the Branin test function; (b) contours of a DACE response surface based on the 21 sampled points shown as dots; (c) a quadratic surface fit to the 21 points; (d) a thin-plate spline fit to the 21 points.

I.IV.I Sequential Methods

- Sequential Parameter Optimization (SPO)
- 7 step process, rather than an automatic algorithm.
- Suggests use of DoE and DACE models.
- Focus on Standardization
- Restrictions
  - Continuous variables
  - Single problem instances

I.IV.ActiveConfigurator

- Based on EGO
- Integrates time bounds
- Incumbent
- Can deal with all parameter types
  - Random forests for categorial parameters

I.IV.Racing and Sharpening

- Both techniques to boost performance
- Racing does it by direct competition of parameter settings (as in Tournament Selection)
- Sharpening increases amount of sample points for promising regions.

I.IV.Parallel Methods: GGA

- Gender based Genetic Algorithm
- 2 genders: only one assessed (win ½ time)
- Based on the idea that mating choice has a higher impact on evolution than natural selection.
- Continous, discrete and categorial variables.

I.IV.Conclusion

- Model-Based approaches take over.
- Very few comparisons so far!
- GGA and ActiveComparator perform competitive on SAT problem.
  - Improve default values significantly!
  - With an advantage for ActiveComparator so far.
  - GGA parallel execution would boost it.
- Much left to do:
  - Standard Toolkit (simple to apply)
  - Decimal, Categorial parameters

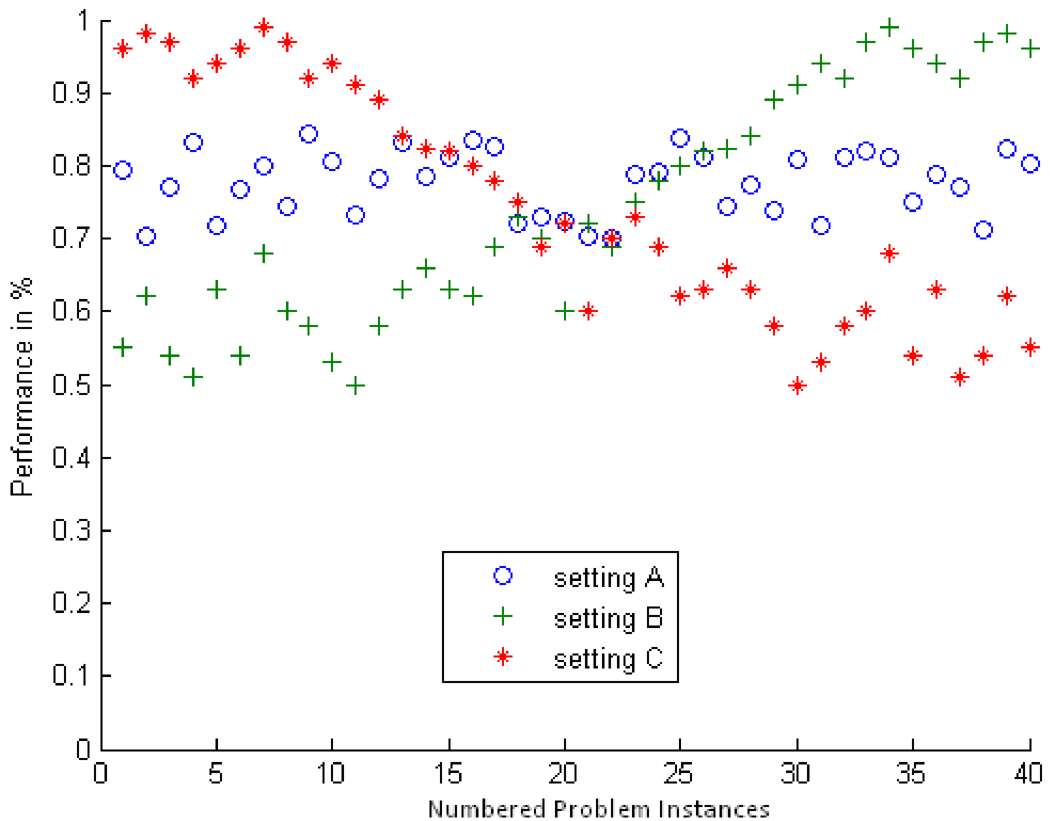
Questions so far!?

Any comments!?!

II.IV. Robust Parameter Settings: Overview

I. Robust Parameters
II. Experiments
III. Results
IV. Consequences

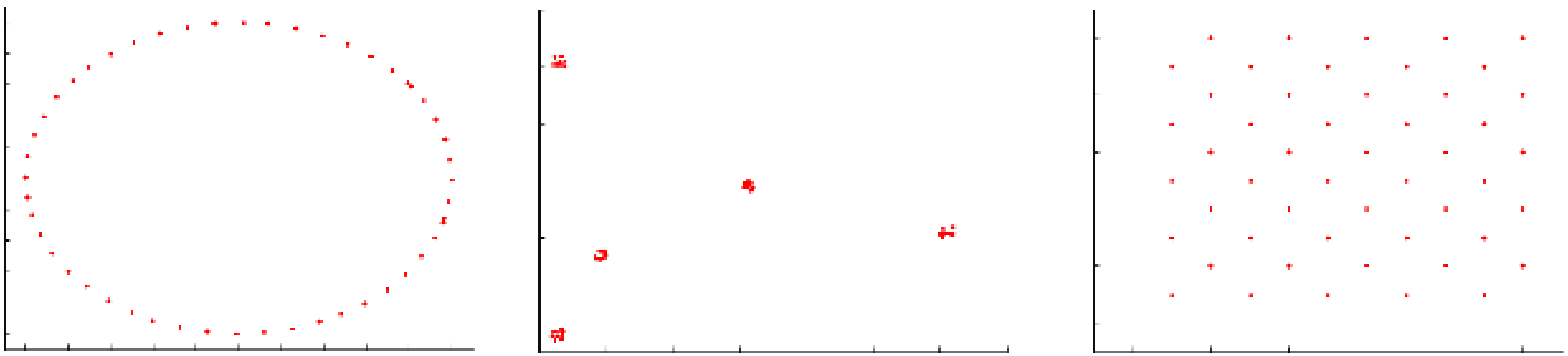
II.I. Robust Parameter Values: The Problem



"Optimization is actually just the opposite of robustness.“ (Marczyk, 2000)

Mittuniversitetet
MID SWEDEN UNIVERSITY

II.II TSP: Problem Instances



circle50 colonies50 uniform50

II.II. Binary PSO: Parameters to be tuned

factor	data type	min	max
iterations	integer	100	500
population size	integer	50	150
α	decimal	0,50	0,99
β	decimal	0,50	0,99

Robust setting used for comparison by Cunkas and Ozsaglam, 2009:

setting	population size	iterations	α	β
robust	150	10.000	0,99	0,99

II.II. Experimental Design

- Full Factorial Design with 3 center points ($2^4 + 3 = 19$)
- 5 repetitions ($19 * 5 = 95$)
- Results compared based by average values
- Matlab implementation of Binary PSO.

II.III. Experimental Results

map	category	parameter setting pop	iter	α	β	solution quality	exec. time (s)	*qdtb* (%)	*etdtb* (%)
circle50	best found	150	500	0.50	0.99	437.33	511	0.0	0.0
	robust	150	500	0.99	0.99	468.65	607	-6.7	-15.8
	fastest	50	100	0.50	0.99	481.47	35	-9.2	1460.0
	center	100	300	0.75	0.75	465.43	239	-6.0	213.8
	best qual/time	50	100	0.50	0.99	481.47	35	-9.2	1460.0
colonies50	best found	150	500	0.50	0.50	498.04	523	0.0	0.0
	robust	150	500	0.99	0.99	541.71	654	-8.1	-20.0
	fastest	50	100	0.50	0.99	580.72	36	-16.3	1452.8
	center	100	300	0.75	0.75	533.88	235	-6.7	222.6
	best qual/time	50	100	0.50	0.99	580.72	36	-14.2	1452.8
uniform50	best found	150	500	0.99	0.99	194.77	634	0.0	0.0
	robust	150	500	0.99	0.99	194.77	634	0.0	0.0
	fastest	50	100	0.99	0.50	214.71	36	-9.3	1761.1
	center	100	300	0.75	0.75	203.86	238	-4.5	266.4
	best qual/time	50	100	0.99	0.50	214.71	36	-9.3	1761.1

qdtb: Quality difference to best found

edtb: Execution time difference to best found

II.IV Experimental Results

- The robust setting was outperformed in 2/3 cases.
- Best settings were different for all the three maps.
- Considering quality/time trade-off, robust setting shown to be very bad.
- Just a DoE screeing process: Much space for improvement on robust setting!

II. Conclusion

- Robust parameter settings
  - Are not necessarily the best choice
  - Can imply weaknesses in quality/time trade-off
- But is a meta model worth the effort? (Expensive experimental preprocessing)
  - Depends on problem complexity and its generalization features.

Now, a question!?

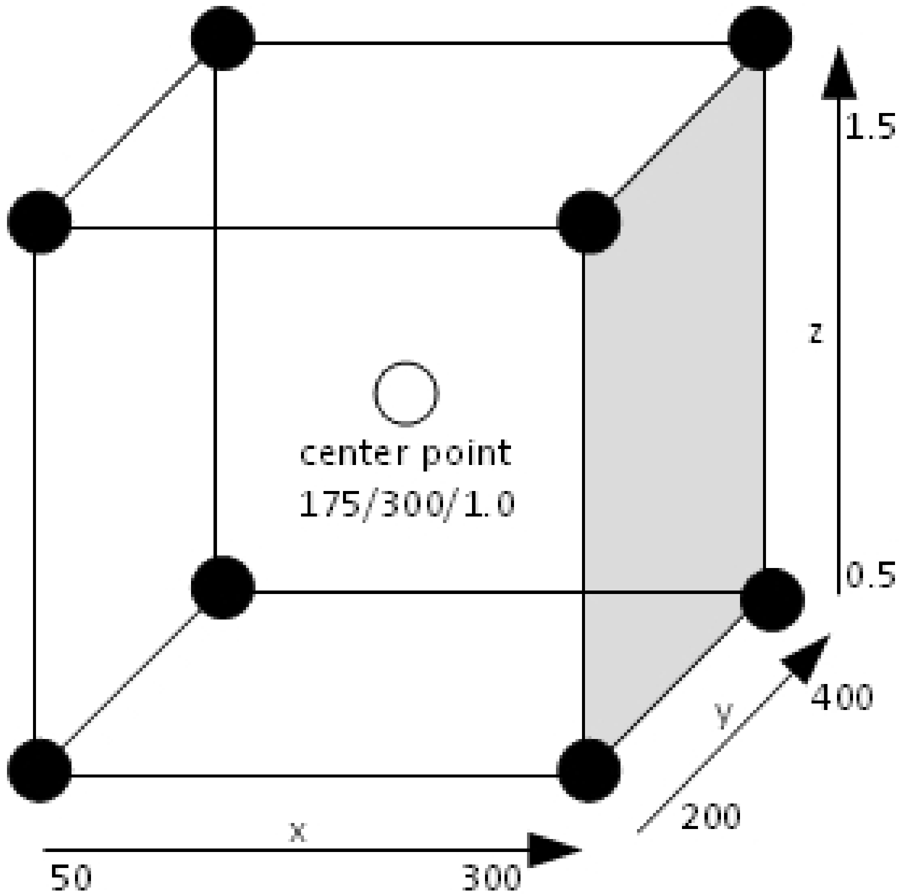
…

or a Comment?

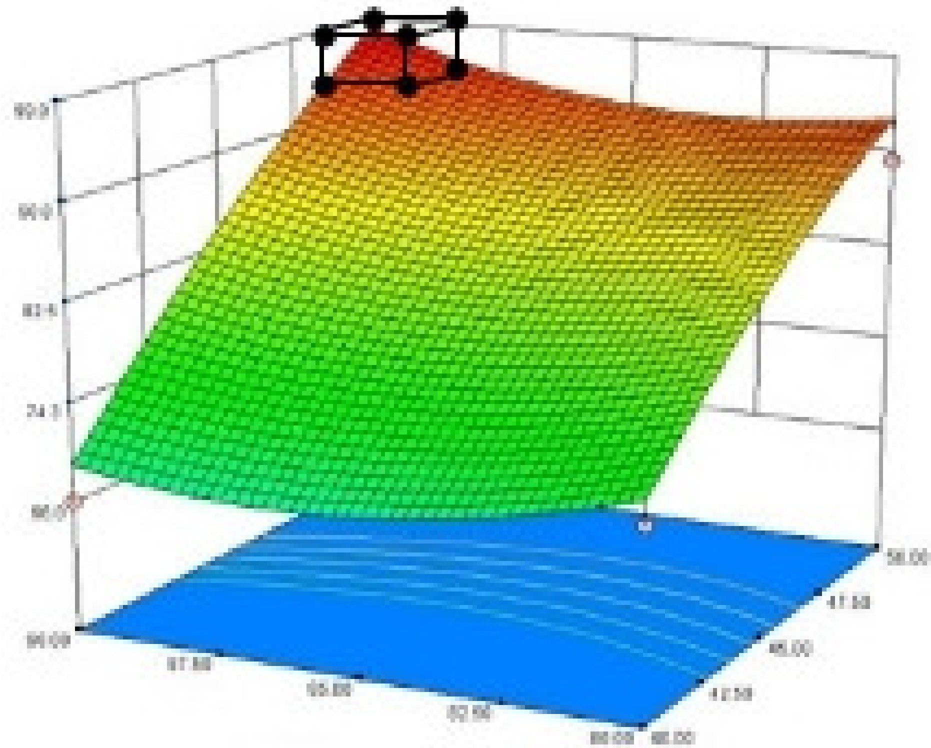
III. Proposed Methodology: Overview

- Introduction
  - Design of Experiments (DoE)
- The Framework
- Conclusions
- Future Work

Design of Experiments



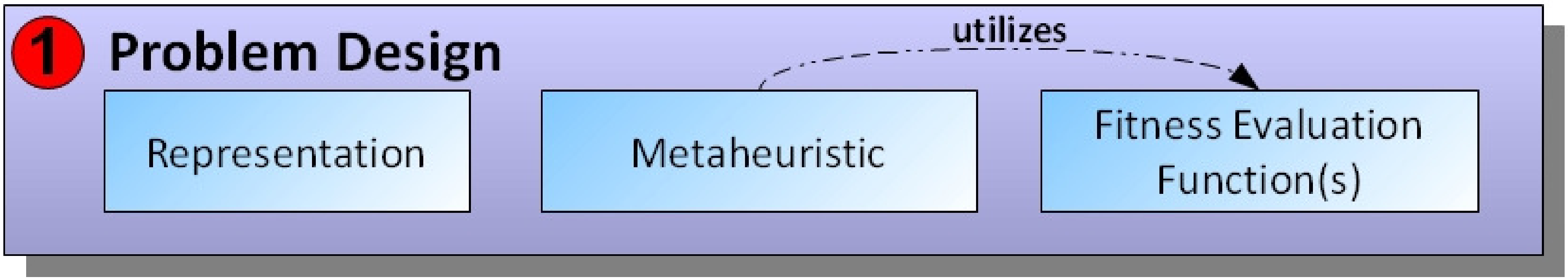
Design of Experiments



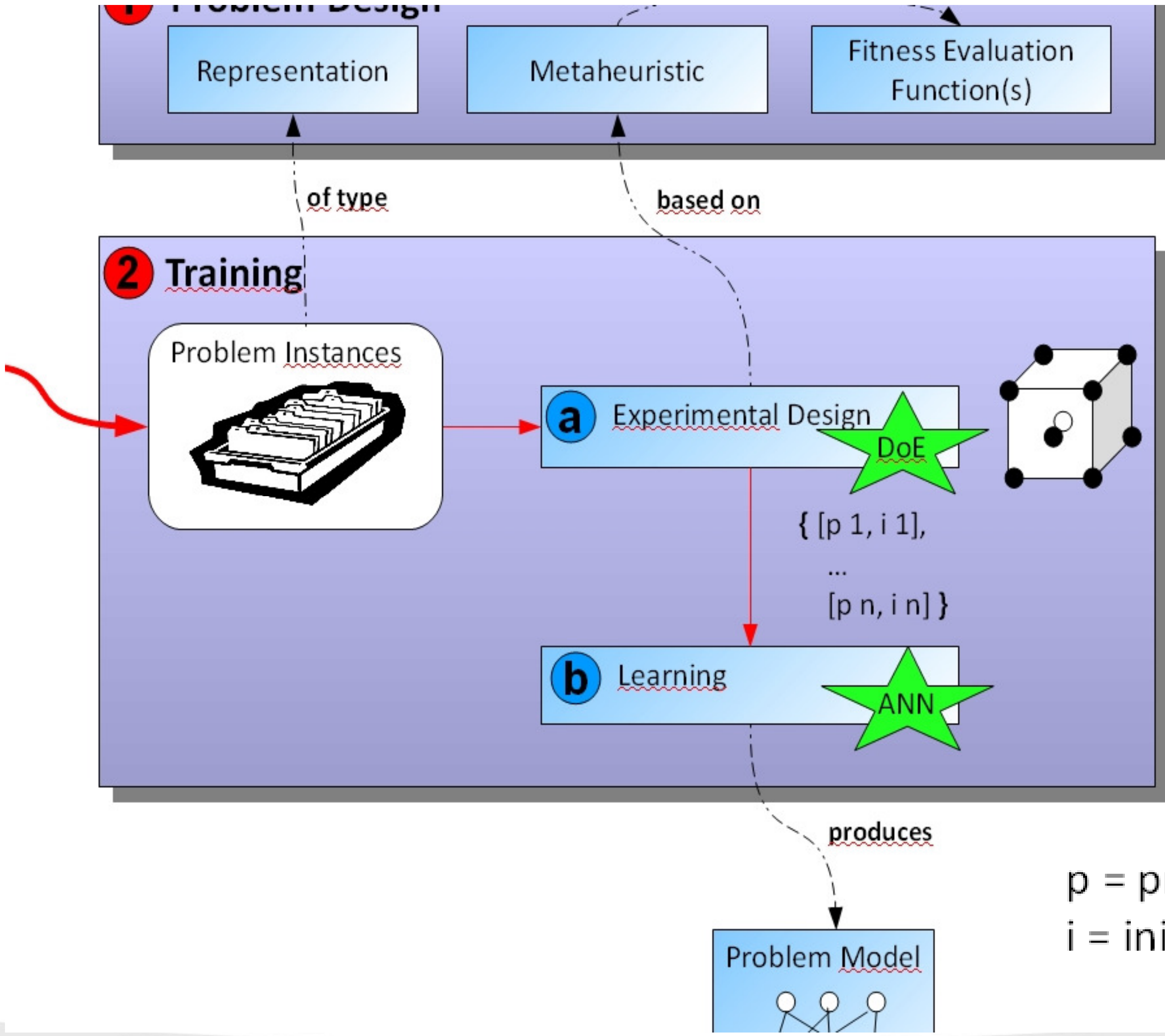
The Framework: Overview

1. Problem Design
2. Training
3. Parameter Retrieval
4. Execution

The Framework, phase 1: Problem Design

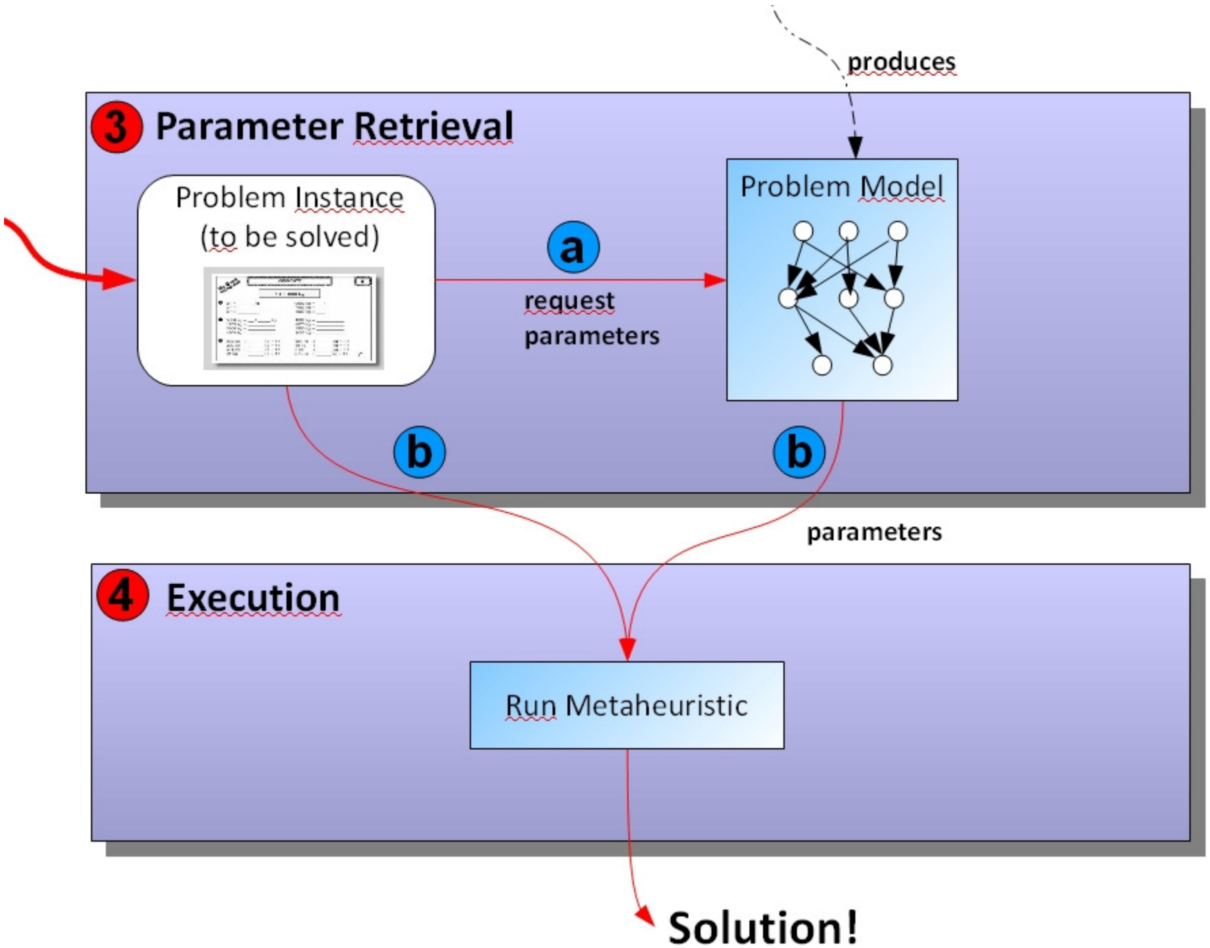


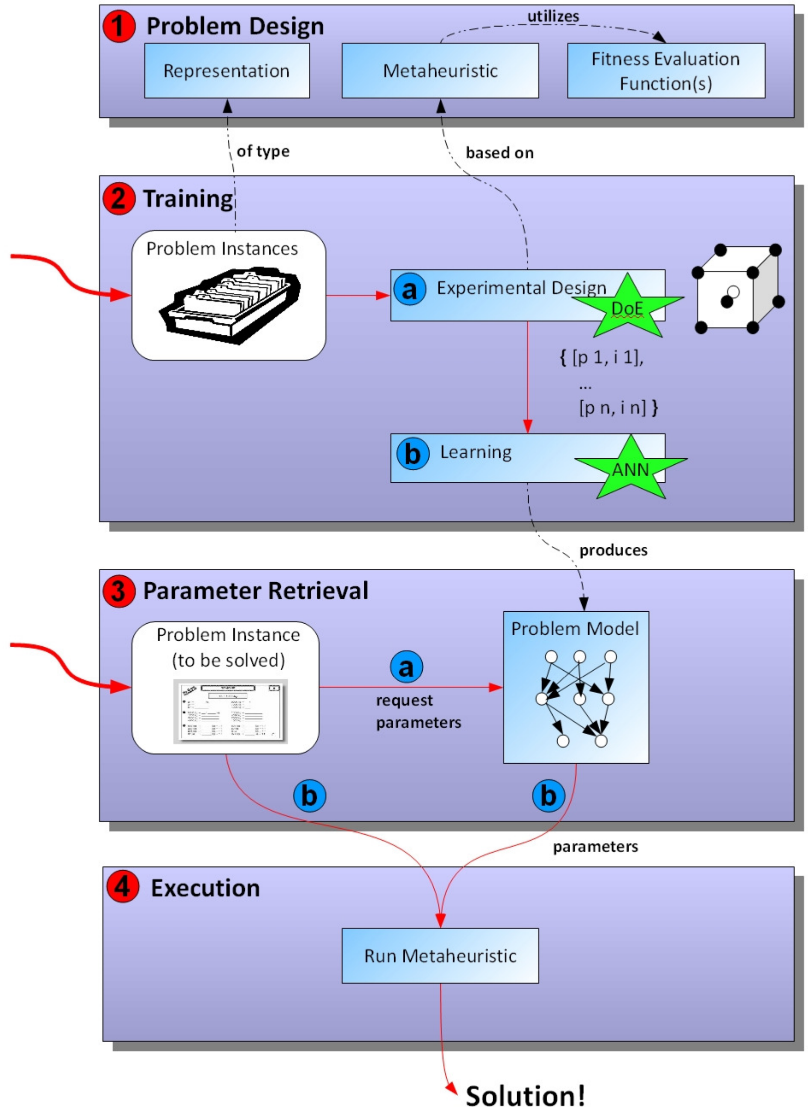
The Framework, phase 2: Training



p = problem instance
i = initial parameter values

The Framework, phases 3+4





p = problem instance
i = initial parameter values

Final Conclusions

- A case for Automatic Parameter Tuning was made.
- DoE might not be the best solution:
  - Models suffer from predictive power
  - Rather DACE for random algorithms for the framework?
- To be improved:
  - Understanding of parameter interactions
  - Optimizing categorial parameter tuning
  - Practical use: common toolkit
  - Take advantage of real Parellelism
    - Evolutionary Algorithms, PSO
    - Parallel statistical learning models

Future Work

- Case studies with ActiveConfigurator
  - extend for parallellism?
- Assess ANNs applicability. Tweaking…
  - initial weighting,
  - cost function…
- Investigating Swarm-based algorithms
  - such as PSO
  - fast execution times, good convergence
  - Parallel by default

Thank you for the attention!

Now, a question!?

A comment maybe?