Automatic Configuration and Learning for Evolutionary Computation

IDEAL 2019, 15.11.2019

Thomas Bäck, Leiden Institute of Advanced Computer Science

Universiteit Leiden
The Netherlands

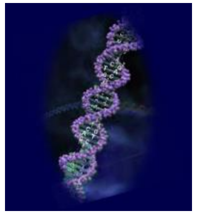
Overview

1.1

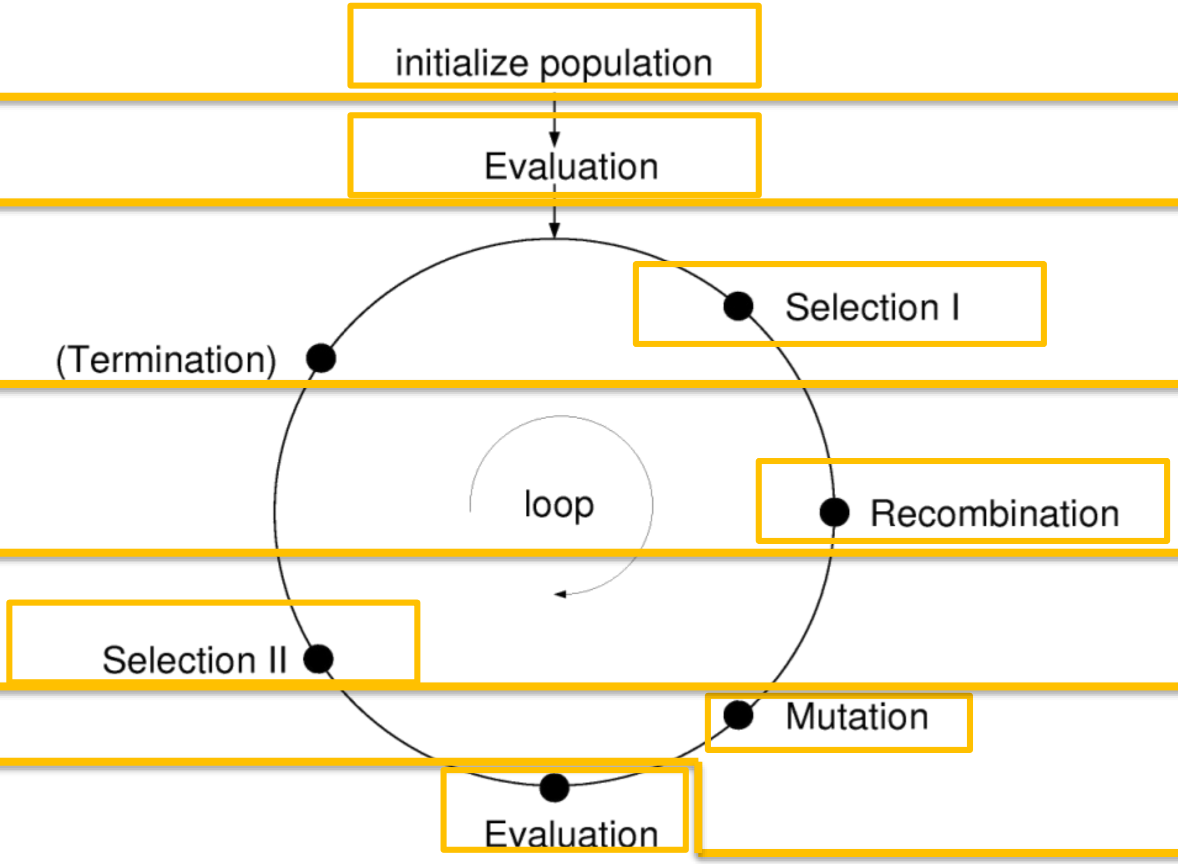
INTRODUCTION TO MY WORLD

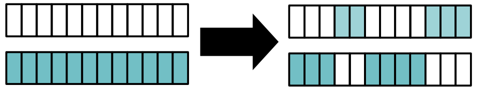
Evolutionary Computation

Huge range of applications: Machine learning, search, engineering, logistics, science, etc.



- Global, direct search algorithms



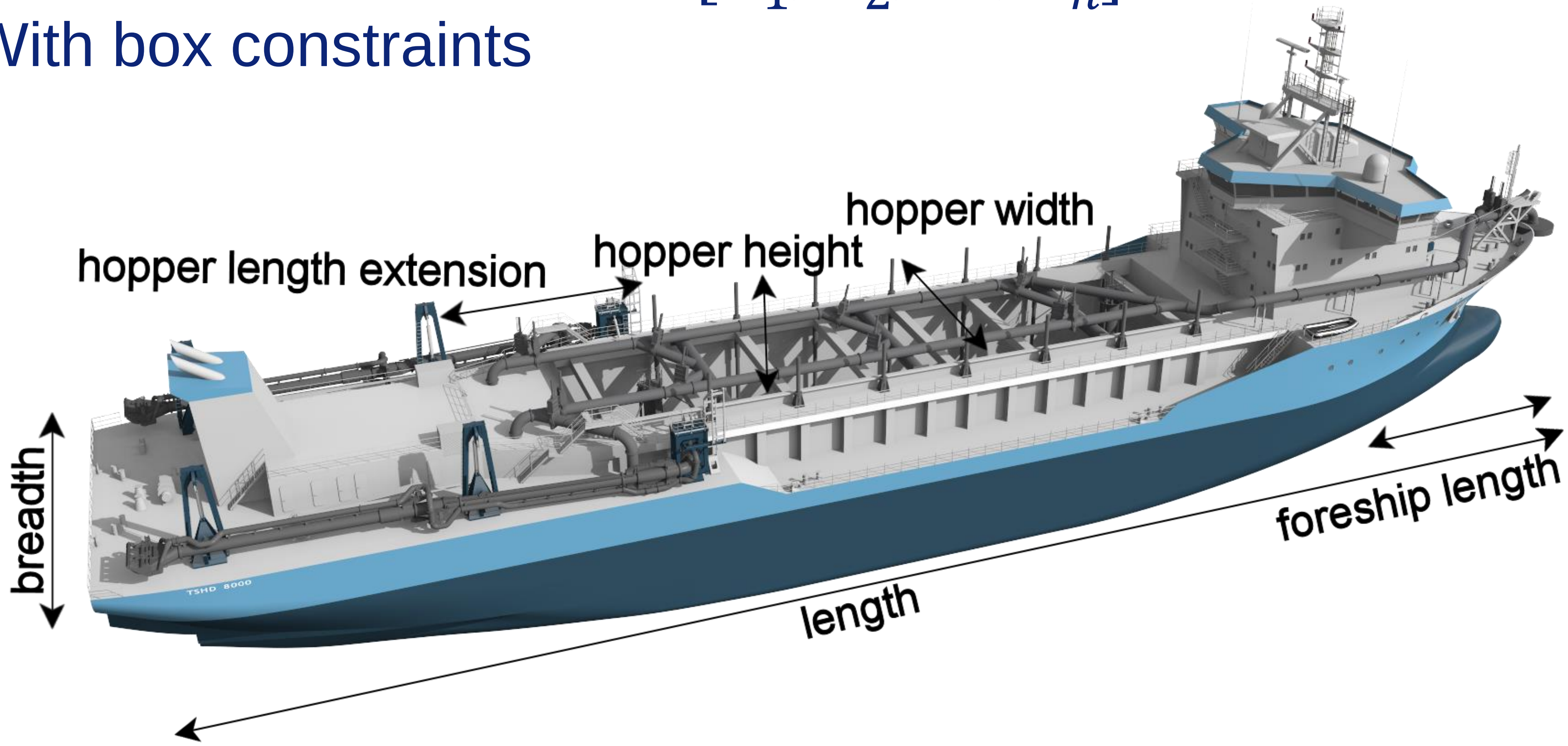
Genetic Algorithm		Evolution Strategy
$\vec{a} \in \{0,1\}^n$	Random	$\vec{a} \in \mathbb{R}^n$
	Objective function:	$f(\vec{a}) \rightarrow max$
Fitness-proportional $p_i = \frac{f_i}{\sum_{j=1}^{\mu} f_j}$		Random $p_i = 1/\mu$
		
None		Deterministic $(\mu,\lambda), (\mu+\lambda)$
iid bit flips w.p. $p \in]0,1[$		iid 1-dim. / n-dim. Gaussian $X_i \sim N(0,\sigma^2)$ $\boldsymbol{X} \sim \boldsymbol{N}(\boldsymbol{0},\boldsymbol{\Sigma})$
	Objective function:	$f(\vec{a}) \rightarrow max$

1.2

APPLICATION EXAMPLES

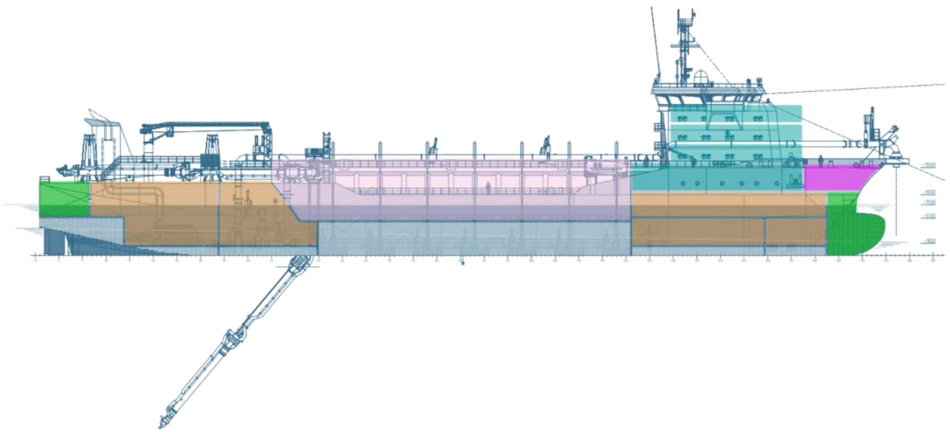
Multiobjective Ship Design: Dredger

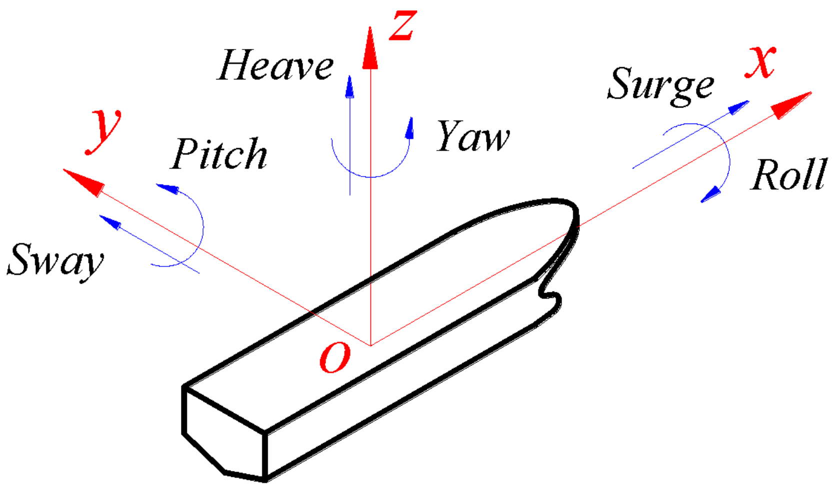
- Decision variables : $\vec{X} = [x_1, x_2, \ldots, x_n]$
  - With box constraints



Dredger ship case

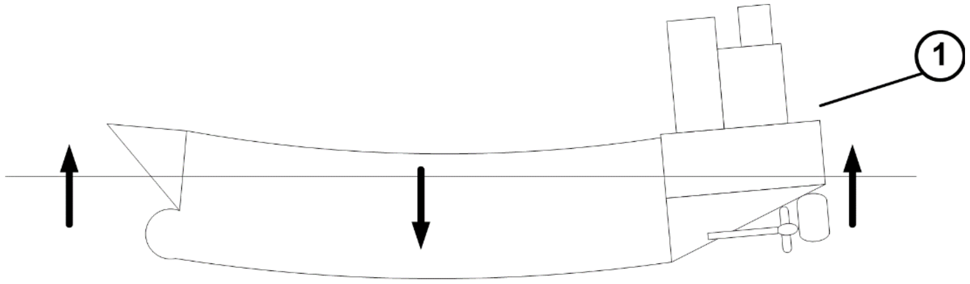
- Constraints : $G_j\left(\vec{X}\right) \leq 0 \qquad \forall_j \in \{1, \ldots, m\}$
  - Space reservation: payload, fueltank, engine, dredging pump, accommodation
  - Regulating authorities: stability, strength, trim, heel

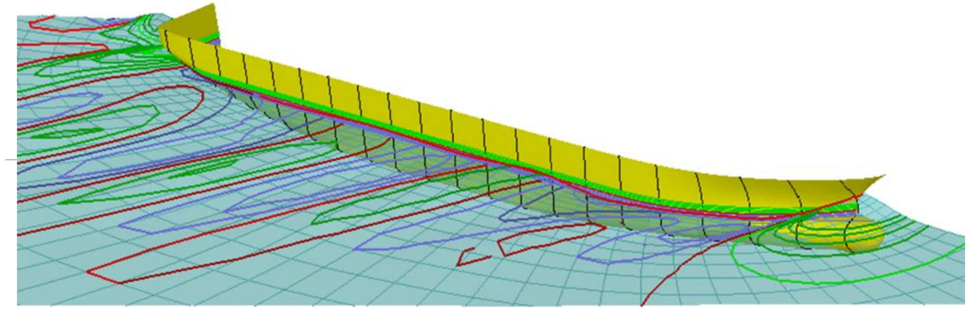


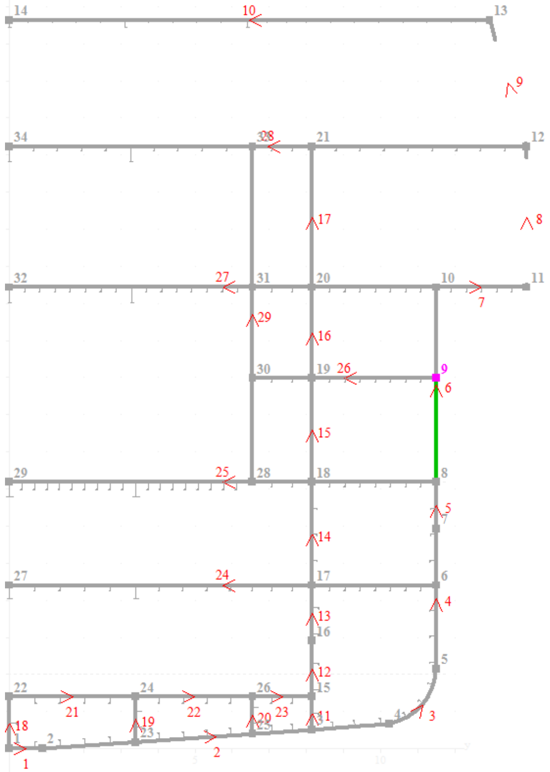


Dredger ship case

- Objectives: $F\left(\vec{X}\right) = [f_1\left(\vec{X}\right), f_2\left(\vec{X}\right), \dots, f_k(\vec{X})]$
  - Minimize: steel weight / building costs
  - Minimize: resistance / operational expenses





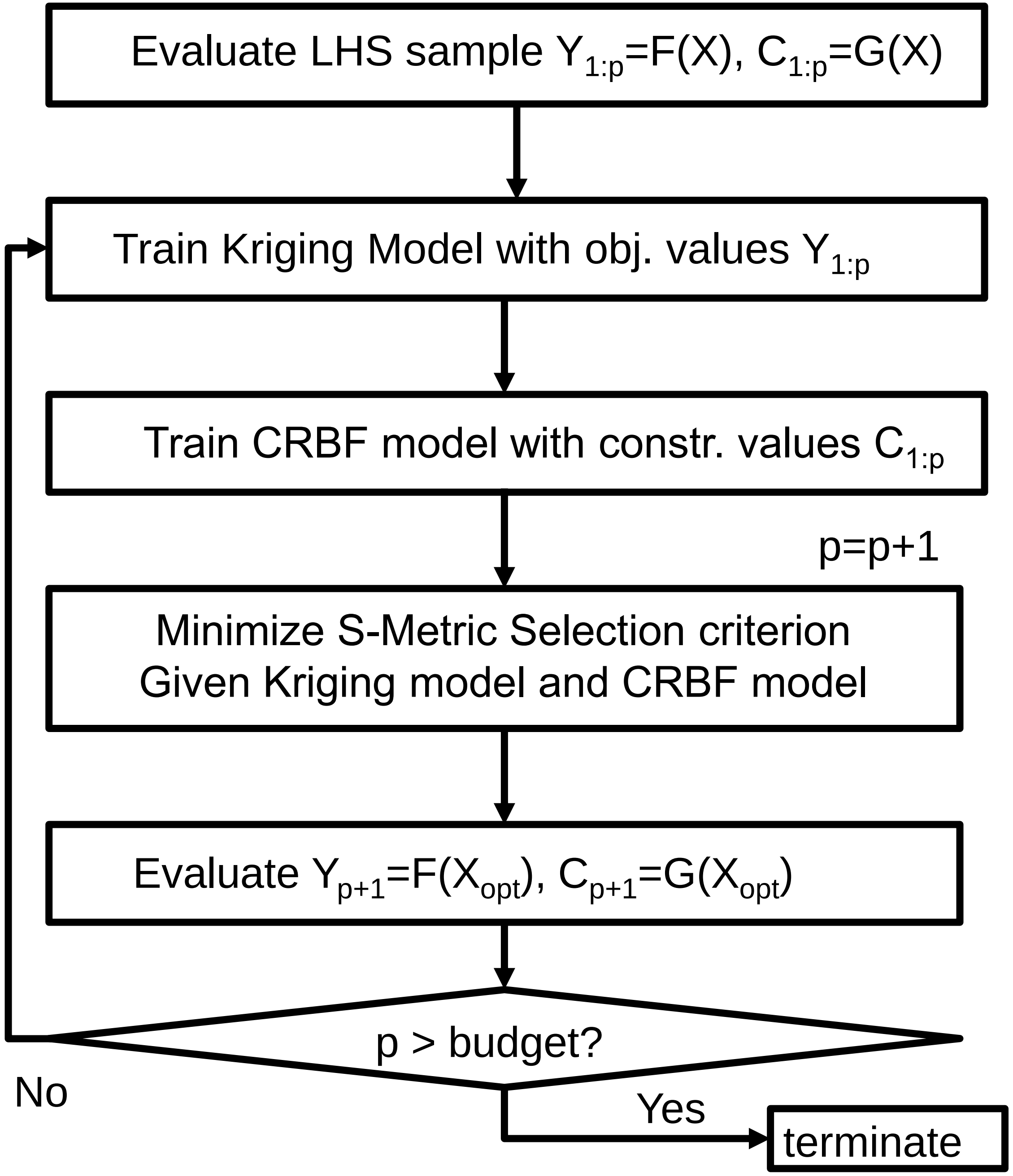


CEGO

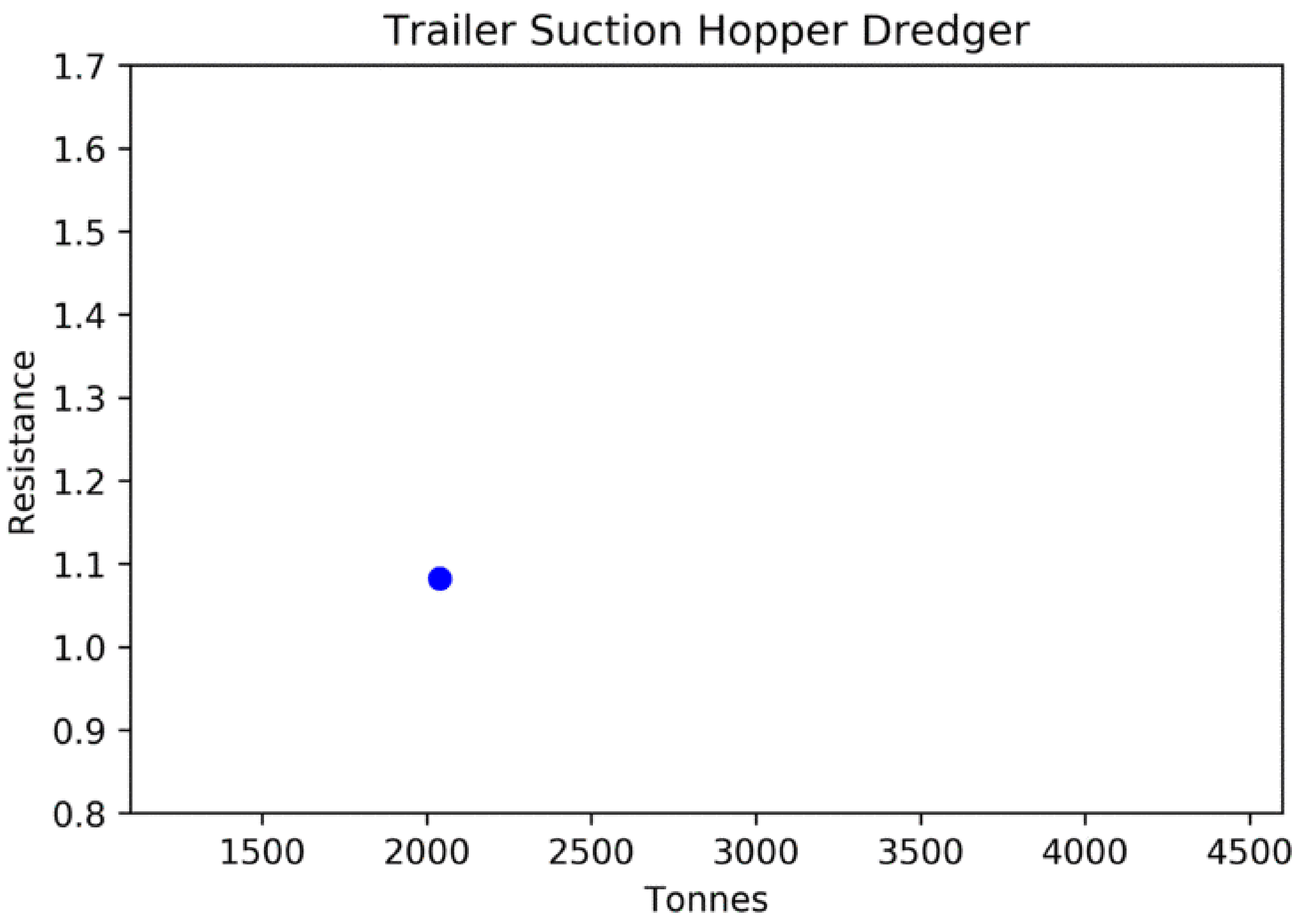
R. DE WINTER, B. VAN STEIN, M. DIJKMAN, AND T. BÄCK (2018). Designing Ships using Constrained Multi-Point Efficient Global Optimization. *Machine Learning, Optimization, and Data Science Conference*, 2018.

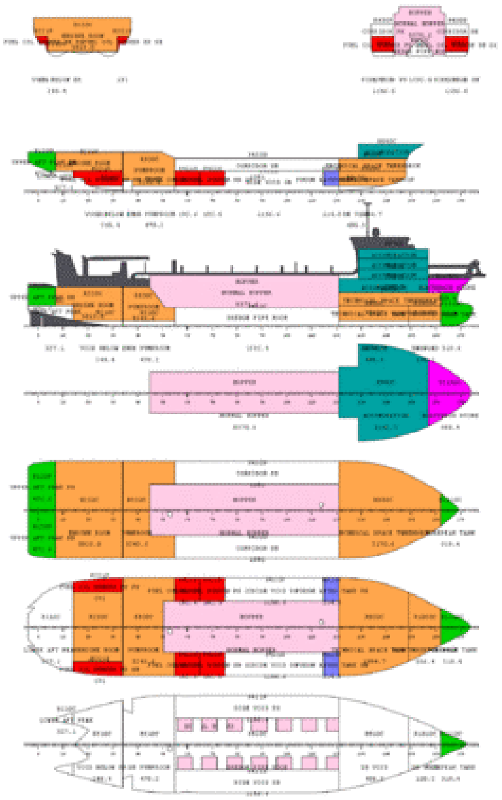
- EGO + Constraint handling + Multiple objectives
- Objectives: Gaussian process models
- Constraints: RBF models
- Combines SACOBRA + SMS-EMOA

S. BAGHERI, W. KONEN, M. EMMERICH, **TH. BÄCK**: Self-adjusting parameter control for surrogate-assisted constrained optimization under limited budgets. *Applied Soft Computing* **61,** 377-393, 2017.



CEGO 200 iterations Dredger

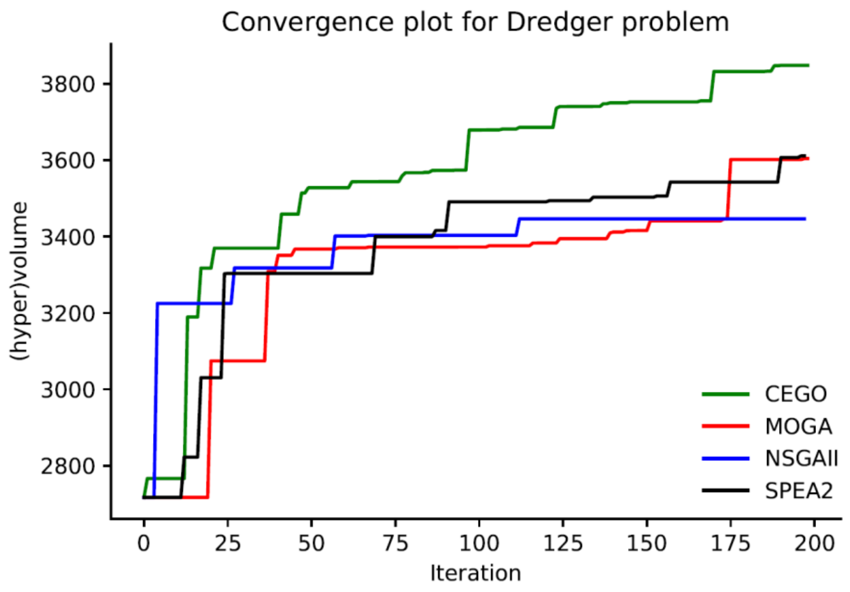




Algorithm Comparison

Pareto Frontier

Convergence Plot



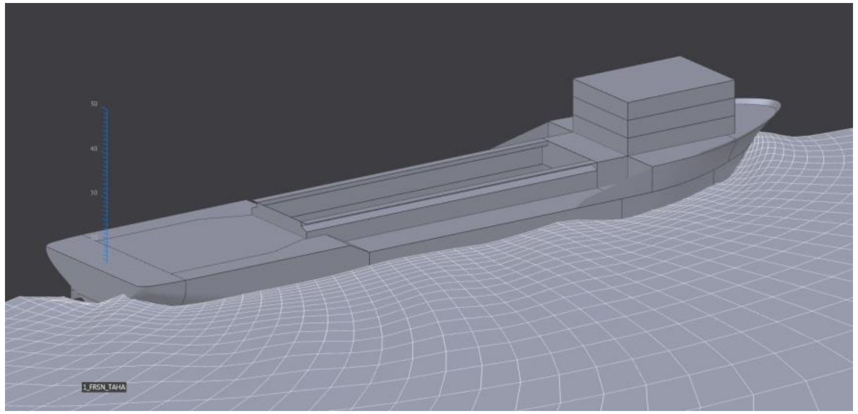
Marked individuals

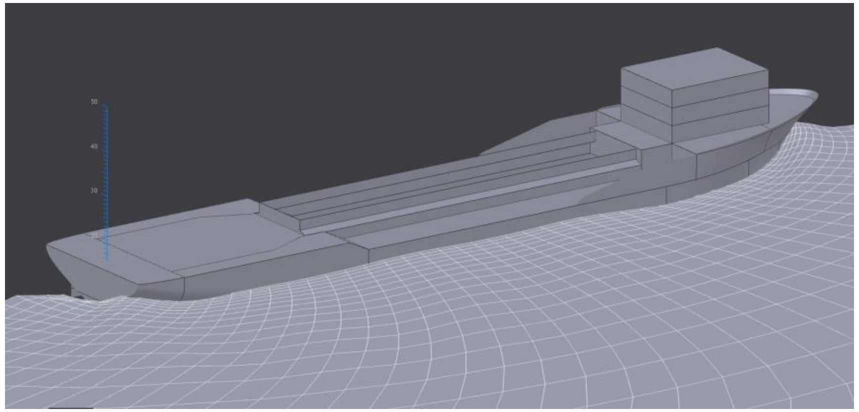
Original design

- Resistance coef: 1.08
- Steel weight: 2039Tonnes

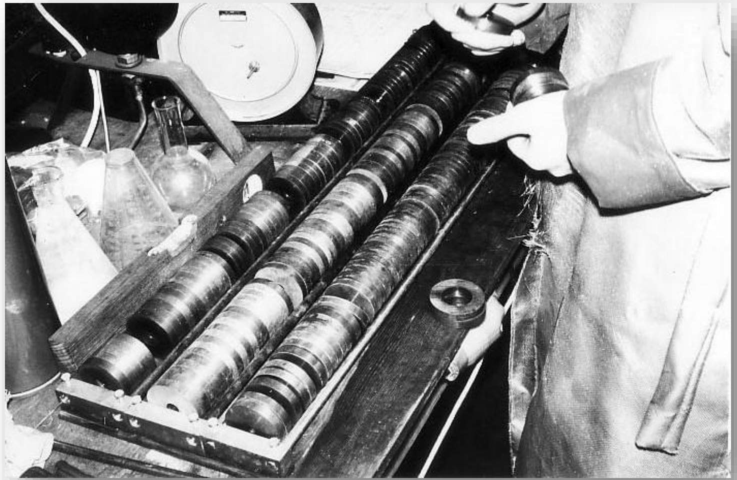
Marked CEGO result

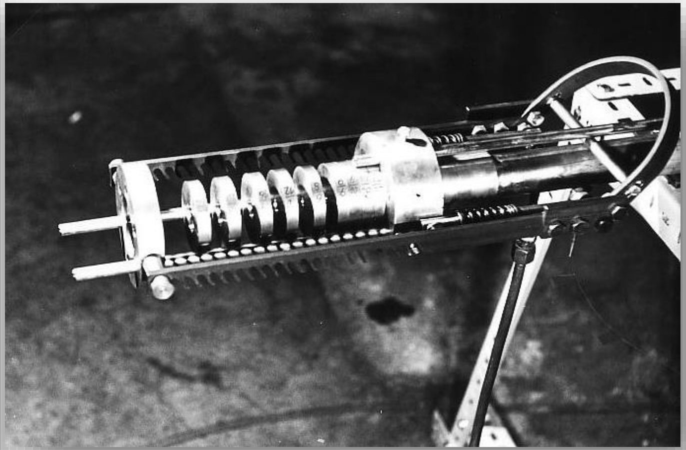
- Resistance coef: 0.877
- Steel weight: 1748Tonnes



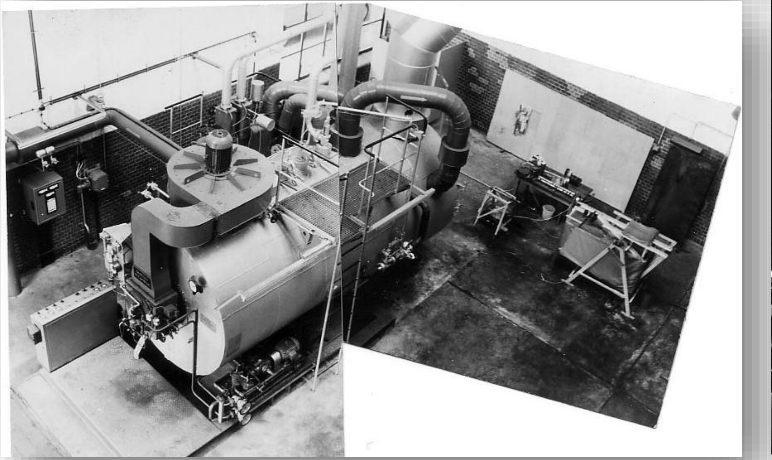


What if … You Had Very Few Trials?

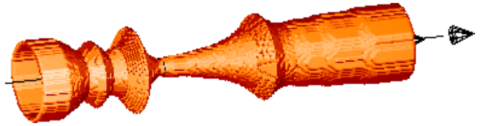




All photos courtesy of Hans-Paul Schwefel.



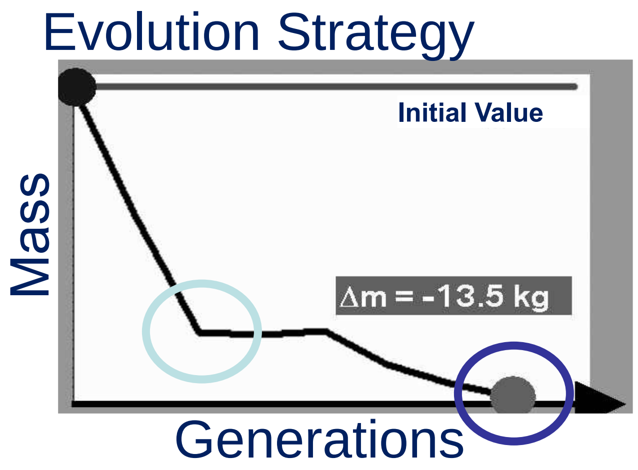
255 experiments

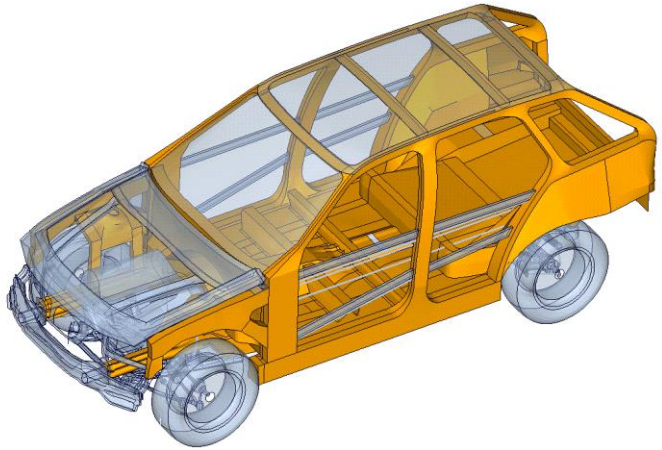


32% efficieny improvement!

J. Klockgether and H.-P. Schwefel: Two-phase nozzle and hollow core jet experiments. In *Proc. 11th Symp. Engineering Aspects of Magnetohydrodynamcis*. Ed. D. Elliott, pp. 141-148. California Institute of Technology, Pasadena, CA, 1970.

MDO Production Runs

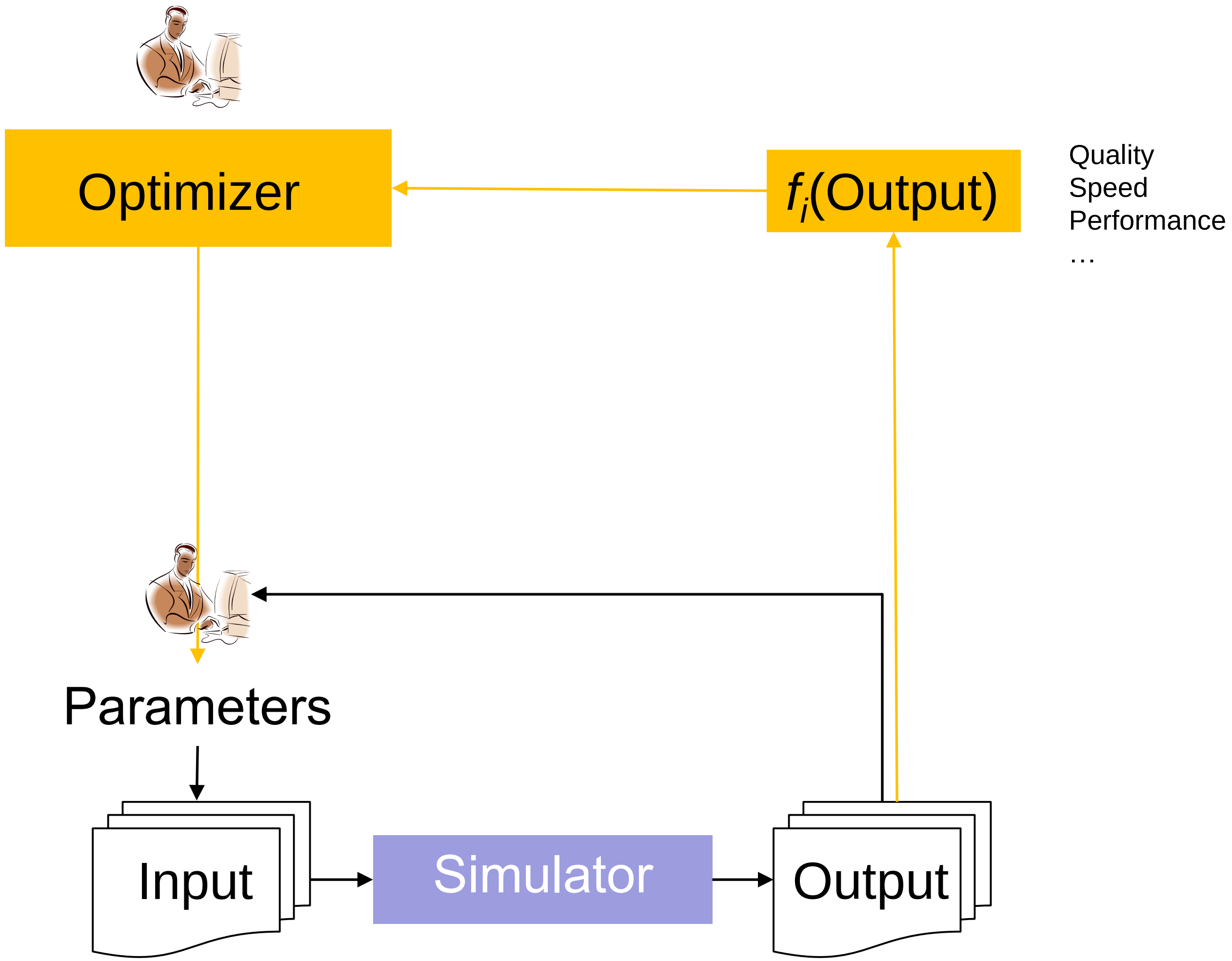




- 13,5 kg weight reduction by advanced ES.
- Beats best so far method significantly.
- Typically faster convergence velocity of ES.
- Development time: 5 wk → 2 wk
- Allows for process integration.
- Still potential for further improvement after 180 shots.

F. Duddeck: Multidisciplinary Optimization of Car Bodies. *Structural and Multidisciplinary Optimization* 35(4), 375-389, 2008.

Optimization – Simulation Loop



2.1

SELF-ADAPTING OPTIMIZATION ALGORITHMS

What Users Want …

- Solve high-dimensional problems
- Very few function evaluations (often < 1,000)
- Many (nonlinear) output constraints
- 24h per function evaluation
- Ideally:
  - part of the workflow
  - integrated, „click the button“ or fully automatic
- We call this „self-adapting“.

Simple Self-Adaptation – one σ

- One σ controls mutation for all x_i
- Mutation: N(0, σ)

$$\vec{a} = ((x_1,\ldots,x_n),\sigma)$$

$$\vec{a}' = ((x_1',\ldots,x_n'),\sigma')$$

$$\sigma' = \sigma \cdot \exp(\tau_0 \cdot N(0,1))$$

$$x_i' = x_i + \sigma' \cdot N_i(0,1)$$

Individual before mutation

Individual after mutation

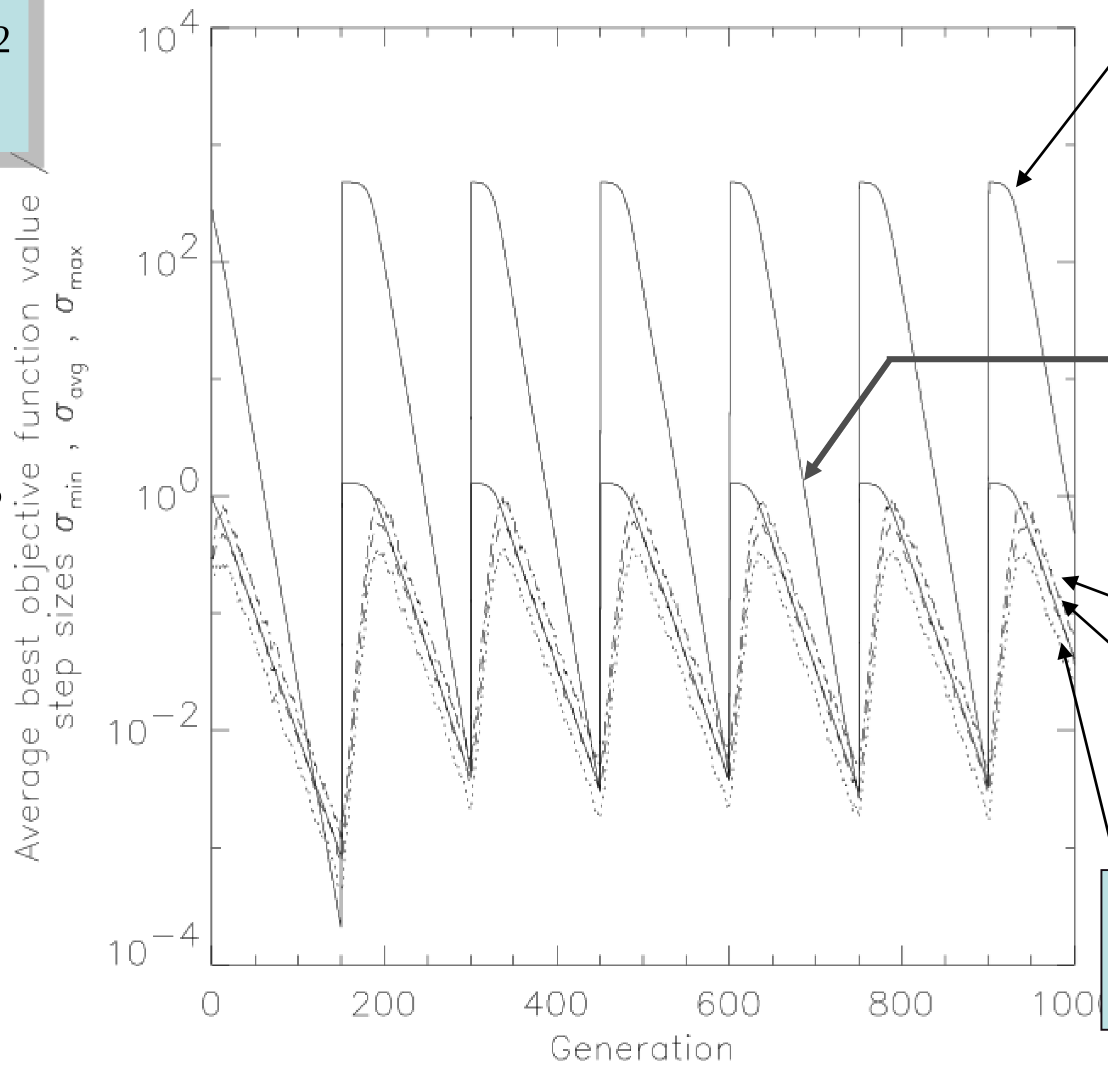
1.: Mutation of step sizes

2.: Mutation of objective variables

Self-adaptation: Example

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

- Dynamic
- Changes every 150 generations



Objective function value

According to theory of optimal step sizes

$$\sigma_{opt} = c_{\mu,\lambda} \frac{R}{n}$$

Largest …

average …

… and smallest step size measured in the population

2.2

STATE OF THE ART: CMA-ES

CMA-Evolution Strategies

- Covariance matrix adaptation ES

Input: $m \in \mathbb{R}^n$, $\sigma \in \mathbb{R}_+$, λ

Initialize: $\mathbf{C} = \mathbf{I}$, and $p_c = \mathbf{0}$, $p_\sigma = \mathbf{0}$,

Set: $c_c \approx 4/n$, $c_\sigma \approx 4/n$, $c_1 \approx 2/n^2$, $c_\mu \approx \mu_w/n^2$, $c_1 + c_\mu \leq 1$, $d_\sigma \approx 1 + \sqrt{\frac{\mu_w}{n}}$,
and $w_{i=1\ldots\lambda}$ such that $\mu_w = \frac{1}{\sum_{i=1}^{\mu} {w_i}^2} \approx 0.3\,\lambda$

While not terminate

$$x_i = m + \sigma y_i, \quad y_i \sim \mathcal{N}_i(\mathbf{0}, \mathbf{C}), \quad \text{for } i = 1, \ldots, \lambda \qquad \text{sampling}$$

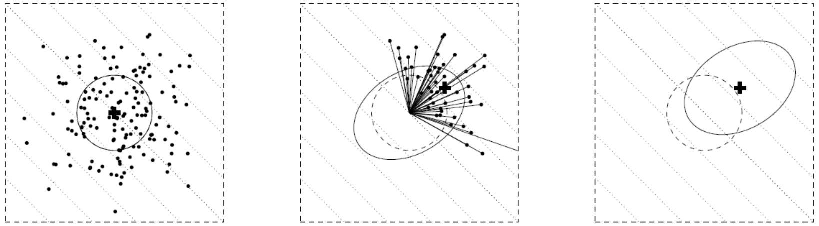
$$m \leftarrow \sum_{i=1}^{\mu} w_i x_{i:\lambda} = m + \sigma y_w \quad \text{where } y_w = \sum_{i=1}^{\mu} w_i y_{i:\lambda} \qquad \text{update mean}$$

$$p_c \leftarrow (1 - c_c)\, p_c + \mathbb{1}_{\{\|p_\sigma\| < 1.5\sqrt{n}\}} \sqrt{1 - (1 - c_c)^2} \sqrt{\mu_w}\, y_w \qquad \text{cumulation for } \mathbf{C}$$

$$p_\sigma \leftarrow (1 - c_\sigma)\, p_\sigma + \sqrt{1 - (1 - c_\sigma)^2} \sqrt{\mu_w}\, \mathbf{C}^{-\frac{1}{2}} y_w \qquad \text{cumulation for } \sigma$$

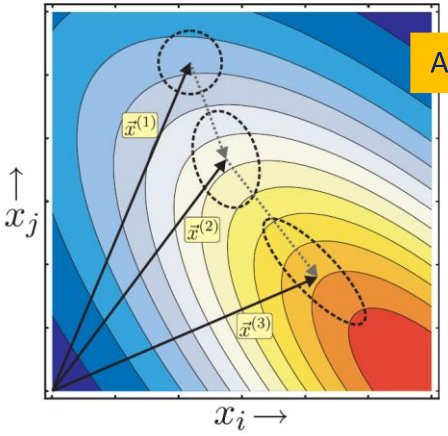
$$\mathbf{C} \leftarrow (1 - c_1 - c_\mu)\, \mathbf{C} + c_1\, p_c {p_c}^{\mathrm{T}} + c_\mu \sum_{i=1}^{\mu} w_i\, y_{i:\lambda} y_{i:\lambda}^{\mathrm{T}} \qquad \text{update } \mathbf{C}$$

$$\sigma \leftarrow \sigma \times \exp\left(\frac{c_\sigma}{d_\sigma}\left(\frac{\|p_\sigma\|}{\mathsf{E}\|\mathcal{N}(\mathbf{0},\mathbf{I})\|} - 1\right)\right) \qquad \text{update of } \sigma$$



Sampling | Estimation | New distrib.

N. Hansen, The CMA-Evolution Strategy: A Tutorial. June 28, 2011. http://www.cmap.polytechnique.fr/~nikolaus.hansen/cmatutorial110628.pdf



Approximate the local Hessian

3.1

OPTIMIZATION FOR ALGORITHM CONFIGURATION

Algorithm Configuration Problem

- Many potential names …
  - Meta-Optimization
  - Meta-Heuristic
  - Hyperparameter optimization
  - Tuning (→ parameters)
  - Calibrating (→ options)
  - Programming by Optimization
    - Levels 0, 1, 2(?)

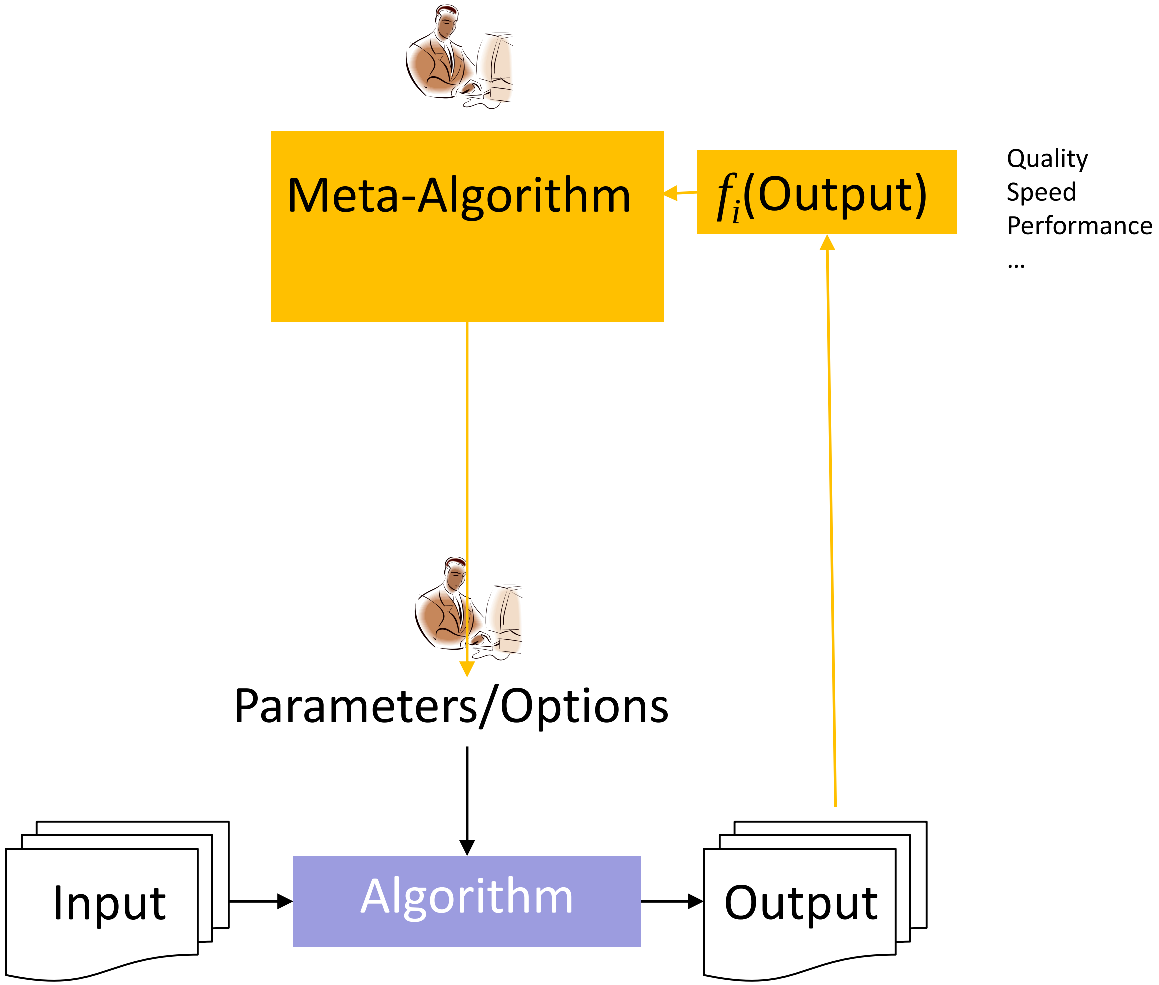
$$\theta^* = \operatorname*{\mathbf{arg\,max}}_{\theta\in\Theta} m\,(\boldsymbol{C}(A(\theta),\Pi))$$

Statistic, e.g., mean

Sample of quality values over problem instances

Algorithm

Set of problem instances



Tuning & Calibrating EAs

- Parameters:
  - Continuous
    - Mutation rate, crossover rate, …
  - Discrete
    - Population sizes, number of XO points, …
  - Categorical
    - Operator types, flags, …

p'_c	p'_m	η'^+	q'	μ'	λ'	z'	s'	r'	e'	Φ
0.202	0.01062	—	—	12	28	—	C	U	1	$2.48201 \cdot 10^1$
0.283	0.09659	—	12	6	16	4	T	S	1	$1.83496 \cdot 10^3$
0.950	0.06581	1.236	—	46	73	—	R	U	0	$3.96824 \cdot 10^3$
0.162	0.06007	—	—	5	14	—	P	D	0	$4.51741 \cdot 10^3$
0.092	0.08388	—	—	3	11	—	C	D	0	$5.66094 \cdot 10^3$
0.606	0.29313	1.066	—	33	98	4	R	S	0	$5.85423 \cdot 10^3$
0.496	0.21327	—	—	9	61	—	C	U	1	$6.24783 \cdot 10^3$
0.880	0.29778	—	—	7	98	—	C	D	0	$6.52485 \cdot 10^3$
0.901	0.24549	—	—	26	84	8	C	S	1	$6.53671 \cdot 10^3$
0.473	0.17765	1.874	—	15	15	—	R	U	1	$7.84848 \cdot 10^3$
0.844	0.17145	—	—	58	74	2	C	S	1	$8.01979 \cdot 10^3$
0.999	0.33714	—	—	11	84	3	P	S	1	$8.03596 \cdot 10^3$
0.227	0.20959	1.469	—	2	12	5	R	S	1	$8.09145 \cdot 10^3$
0.854	0.26478	1.488	—	80	98	—	R	D	1	$8.42532 \cdot 10^3$
0.180	0.24580	—	—	73	81	6	C	S	0	$8.61893 \cdot 10^3$
0.922	0.38617	1.336	—	12	14	—	R	U	1	$8.68986 \cdot 10^3$
0.154	0.28594	1.044	—	38	77	—	R	U	0	$9.06912 \cdot 10^3$
0.152	0.45008	1.031	—	90	90	5	R	S	0	$9.29234 \cdot 10^3$
0.998	0.33453	1.036	—	65	66	—	R	_	0	$9.35776 \cdot 10^3$
0.982	0.22989	—	—	36	52	—	P	_	1	$9.40567 \cdot 10^3$
0.205	0.23960	—	3	7	20	—	T	_	0	$9.41534 \cdot 10^3$
0.595	0.46653	1.247	—	38	40	—	R	U	1	$9.60902 \cdot 10^3$
0.303	0.39338	—	18	38	71	5	T	S	1	$9.69947 \cdot 10^3$
0.764	0.49582	—	—	9	58	—	P	U	1	$1.01754 \cdot 10^4$
0.198	0.49794	1.189	—	22	32	—	R	U	1	$1.11008 \cdot 10^4$
0.814	0.46578	1.265	—	5	11	—	R	_	1	$1.11710 \cdot 10^4$
0.678	0.26395	—	—	5	5	—	C	U	1	$1.14654 \cdot 10^4$
0.230	0.30343	—	12	9	11	—	T	D	0	$1.17944 \cdot 10^4$
0.625	0.37873	—	—	11	30	—	C	_	1	$1.18864 \cdot 10^4$
0.685	0.31320	—	10	7	14	7	T	S	0	$1.36929 \cdot 10^4$

Initial random population

Result …

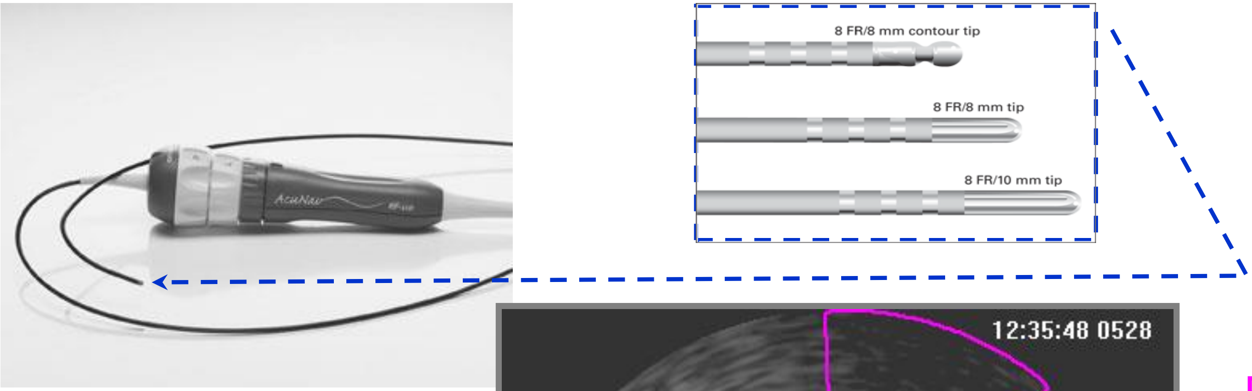
- Seven orders of magnitude improvement:
  - (7,17)-GA
  - Tournament selection, q=20
  - Uniform crossover, p_c=0.421
  - p_m=0.00383

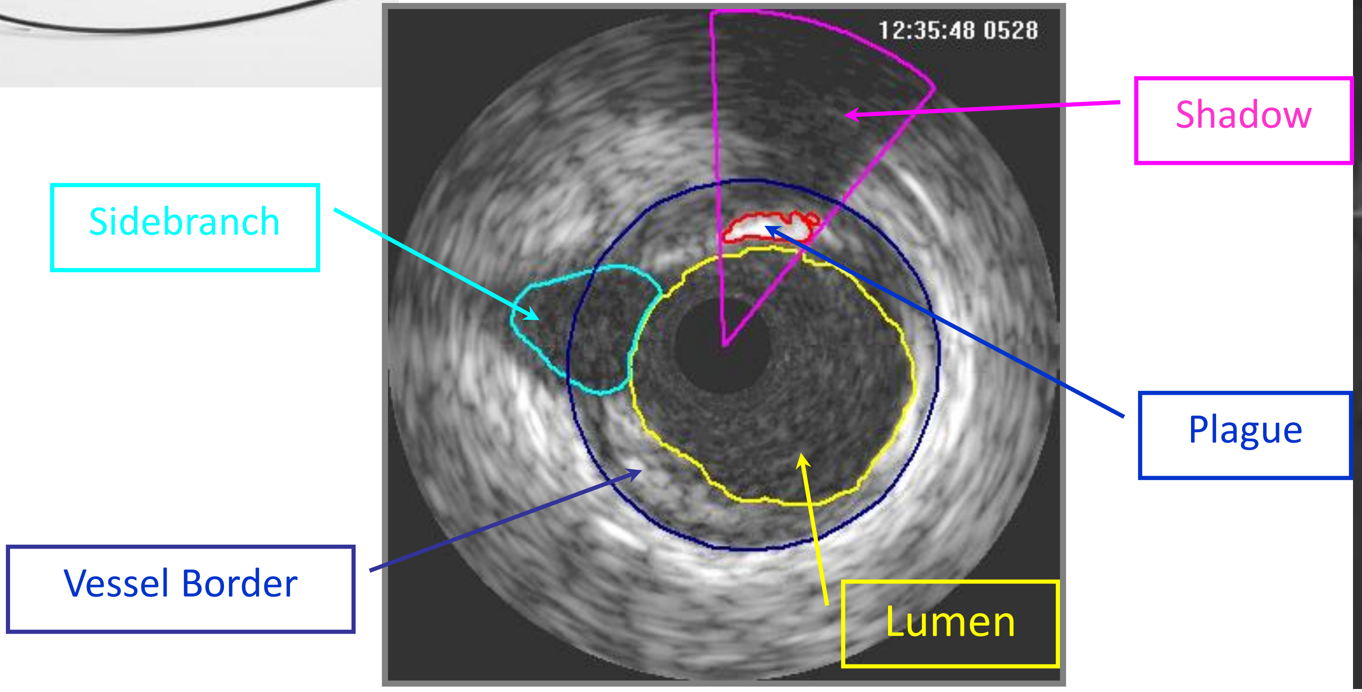
T. Bäck. Parallel optimization of evolutionary algorithms. In Y. Davidor, H.-P. Schwefel, and R. Männer, editors, *Parallel Problem Solving from Nature - PPSN III, International Conference on Evolutionary Computation*, volume 866 of *Lecture Notes in Computer Science*, 418-427. Springer, Berlin, 1994.

p'_c	p'_m	η'^{+}	q'	μ'	λ'	z'	s'	r'	e'	Φ
0.421	0.00383	—	20	7	17	—	T	U	0	$2.164470 \cdot 10^{-6}$
0.421	0.00383	—	20	7	17	—	T	U	0	$2.164470 \cdot 10^{-6}$
0.573	0.00383	—	20	7	17	—	T	U	0	$2.712040 \cdot 10^{-6}$
0.404	0.00383	—	20	7	17	—	T	U	1	$3.473870 \cdot 10^{-6}$
0.570	0.00383	—	20	7	17	—	T	U	0	$3.621860 \cdot 10^{-6}$
0.424	0.00383	—	20	7	17	—	T	D	0	$4.720930 \cdot 10^{-6}$
0.413	0.00383	—	20	7	17	—	T	U	0	$9.608950 \cdot 10^{-6}$
0.414	0.00383	—	20	7	17	—	T	U	0	$1.012210 \cdot 10^{-5}$
0.592	0.00383	—	20	7	19	—	T	U	0	$1.420130 \cdot 10^{-5}$
0.582	0.00383	—	20	7	19	—	T	U	0	$1.420130 \cdot 10^{-5}$
0.418	0.00383	—	20	7	17	—	T	U	0	$1.573790 \cdot 10^{-5}$
0.423	0.00383	—	20	7	17	—	T	U	0	$1.819870 \cdot 10^{-5}$
0.419	0.00383	—	20	7	17	—	T	U	0	$2.168210 \cdot 10^{-5}$
0.417	0.00383	—	20	7	19	—	T	U	0	$2.720590 \cdot 10^{-5}$
0.383	0.00383	—	20	7	17	—	T	U	0	$3.121010 \cdot 10^{-5}$
0.383	0.00383	—	20	7	17	—	T	U	0	$3.249550 \cdot 10^{-5}$
0.571	0.00383	—	20	7	20	—	T	U	0	$4.695690 \cdot 10^{-5}$
0.568	0.00383	—	20	7	17	—	T	U	0	$4.912150 \cdot 10^{-5}$
0.412	0.00383	—	20	7	17	—	T	U	0	$5.726070 \cdot 10^{-5}$
0.416	0.00383	—	20	7	19	—	T	U	0	$7.530270 \cdot 10^{-5}$
0.400	0.00383	—	20	7	20	—	T	U	0	$8.602180 \cdot 10^{-5}$
0.439	0.00383	—	20	7	19	—	T	U	0	$9.137410 \cdot 10^{-5}$
0.415	0.00383	—	20	7	19	—	T	U	0	$9.137410 \cdot 10^{-5}$
0.426	0.00383	—	20	7	17	—	T	U	0	$1.507170 \cdot 10^{-4}$
0.394	0.00383	—	20	7	17	—	T	U	0	$1.610950 \cdot 10^{-4}$
0.593	0.00383	—	20	7	19	—	T	U	0	$7.921950 \cdot 10^{-4}$
0.558	0.00383	—	20	7	17	—	T	U	0	$1.032500 \cdot 10^{-3}$
0.586	0.00383	—	20	7	19	—	T	U	0	$1.736870 \cdot 10^{-3}$
0.424	0.00383	1.330	20	7	17	—	R	U	0	$4.471430 \cdot 10^{-2}$
0.552	0.00383	—	20	7	17	—	M	U	0	$1.720780 \cdot 10^{-1}$

Final population (gen 50)

Example: IVUS Image Classification





R. Li, M.T.M. Emmerich, J. Eggermont, Th. Bäck, M. Schütz, J. Dijkstra, J.H.C. Reiber: Mixed Integer Evolution Strategies for Parameter Optimization. *Evolutionary Computation* **21**(1), 29-64, 2013.

R. Li, M.T.M. Emmerich, J. Eggermont, E.G.P. Bovenkamp, Th. Bäck, J. Dijkstra, J.H.C. Reiber. Mixed-Integer Optimization of Coronary Vessel Image Analysis using Evolution Strategies. In *GECCO 2006: Proceedings of the 7th annual conference on Genetic and Evolutionary Computation, Seattle, WA, Jul. 8-12, 2006,* 1645-1652, ACM Press, NY, 2006.

Results of Algorithm Configuration

- Feature detector parameters:
  - 16 parameters
  - Integer, boolean, categorical
- Performance improvement:
  - Around 40%

name	type	range	dependencies	default
	—			
maxgray	integer	[2, 150]	> mingray	35
mingray	integer	[1, 149]	< maxgray	1
connectivity	ordinal	4,6,8		6
relativeopenings	boolean	{false,true}		true
nrofcloses	integer	[0, 100]	used if not relativeopenings	5
nrofopenings	integer	[0, 100]	used if not relativeopenings	45
scanlinedir	ordinal	{0,1,2}		1
scanindexleft	integer	[-100, 100]	< scanindexright	-55
scanindexright	integer	[-100, 100]	> scanindexleft	7
centermethod	ordinal	{0,1}		1
fitmodel	ordinal	{ellipse, circel}		ellipse
sigma	continuous	[0.5 10.0]		0.8
scantype	ordinal	{0,1,2}		0
sidestep	integer	[0, 20]		3
sidecost	continuous	[0.0, 100]		5
nroflines	integer	[32, 256]		128

	default parameters		parameter solution 1		parameter solution 2		parameter solution 3		parameter solution 4		parameter solution 5	
dataset	fitness	s.d.	fitness	s.d.	fitness	s.d.	fitness	s.d.	fitness	s.d.	fitness	s.d.
1	395.2	86.2	62%	39.5	159.8	43.5	185.4	43.0	144.8	42.0	271.0	74.8
2	400.2	109.	54%	59.2	180.7	58.4	207.2	69.2	232.7	71.0	352.0	73.1
3	344.8	65.6	40%	59.8	203.9	70.1	164.4	49.7	183.9	80.3	327.1	55.9
4	483.1	110.	41%	92.7	269.0	73.2	250.4	80.4	173.2	64.7	330.1	82.2
5	444.2	90.6	17%	00.9	370.9	102.5	462.2	377.3	168.7	64.0	171.8	54.5
			Avg: 43%									

Performance of the best found MI-ES parameter solutions

Mixed-Integer Evolution Strategies

Continuous parameters:

$$
\begin{aligned}
&\textbf{for } i = 1, \ldots, n_r \textbf{ do}\\
&\quad s_i' \leftarrow s_i \exp(\tau_g \mathrm{N}_g + \tau_l \mathrm{N}(0,1))\\
&\quad r_i' = r_i + N(0, s_i')\\
&\textbf{end for}
\end{aligned}
$$

Discrete parameters:

$$
\begin{aligned}
&\textbf{for } i = 1, \ldots, n_z \textbf{ do}\\
&\quad q_i' \leftarrow q_i \exp(\tau_g \mathrm{N}_g + \tau_l \mathrm{N}(0,1))\\
&\quad z_i' \leftarrow z_i + \mathrm{G}(0, q_i')\\
&\textbf{end for}
\end{aligned}
$$

Categorical parameters:

$$
\begin{aligned}
&p_i' := 1/[1 + \tfrac{1-p_i}{p_i} * \exp(-\tau_l * \mathrm{N}(0,1))]\\
&\textbf{for } i \in \{1, \ldots, n_d\} \textbf{ do}\\
&\quad \textbf{if } \mathrm{U}(0,1) < p_i' \textbf{ then}\\
&\quad\quad d_i' \leftarrow \text{uniformly randomly value from } D_i\\
&\quad \textbf{end if}\\
&\textbf{end for}
\end{aligned}
$$

Learning rates (global)

Learning rates (local)

Geometrical distribution

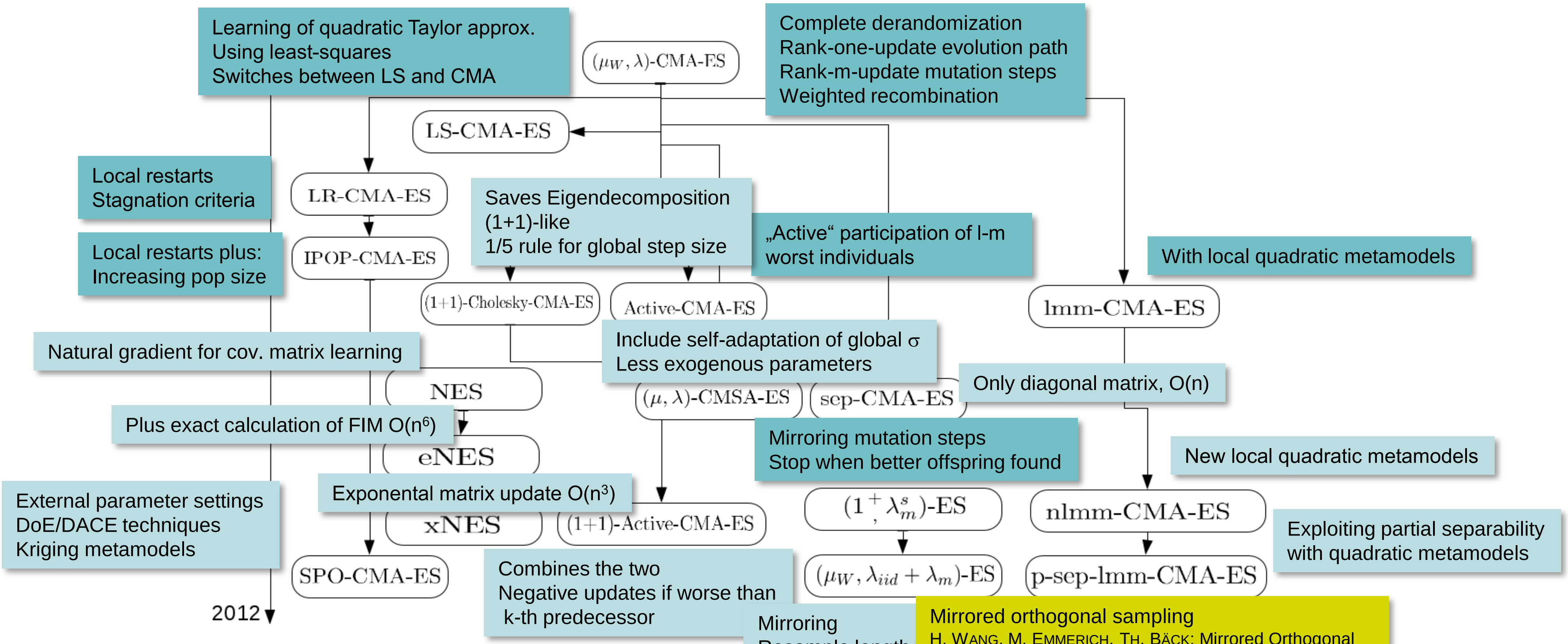
Mutation probabilities

3.2

ALGORITHM CONFIGURATION FOR ES

Contemporary Evolution Strategies

- „Contemporary“ = starting with CMA-ES (1996)
- Taxonomy:



T. Bäck, C. Foussette, P. Krause: *Contemporary Evolution Strategies*.

Mirrored orthogonal sampling
H. Wang, M. Emmerich, Th. Bäck: Mirrored Orthogonal Sampling with Pairwise Selection in Evolution Strategies. In *Proceedings of the 29th Annual ACM Symposium on Applied Computing, SAC ´14, Gyeongju, Korea, March 24-28, 2014*, 154-156. ACM, New York, NY, 2014.

Idea: ES-Superstructure

1.	Active Update:	Use worst individuals too for adapting mutation pdf
2.	Elitism:	($\mu+\lambda$)-selection (no worsening)
3.	Mirrored Sampling:	Sample mirror images as well
4.	Orthogonal Sampling:	Orthogonal mutation vectors
5.	Pairwise Selection:	Prevent mirrored vectors in recombination to avoid cancellation
6.	$1/\mu$ Recombination Weights:	$1/\mu$ recombination weights instead of default
7.	Sequential Selection:	Immediately compare offspring to parents
8.	Threshold Convergence:	Force threshold length of mutation vectors to increase exploration
9.	Two-Point step-size Adaptation:	Two offspring are used to determine whether global σ increase/decrease
10.	Increasing Population (IPOP/BIPOP):	Increasing population size in restarts and alternating
11.	Quasi-Gaussian Sampling (Sobol/Halton):	Quasi-random sequences for random numbers

4,608 possible structures $\{0,1\}^9 \times \{0,1,2\}^2$

Meta-algorithm:

- (1,12)-GA
- self-adaptive mutation
- 240 evaluations (5.2% of search space)

S. van Rijn, H. Wang, M. van Leeuwen, Th. Bäck: Evolving the Structure of Evolution Strategies. In: *IEEE Symposium Series on Computational Intelligence, Athens, Greece, Dec. 6-9, 2016.* IEEE Press, Piscataway, NJ, 2016.

Idea: ES-Superstruct

1. Active Update:
2. Elitism:
3. Mirrored Sampling:
4. Orthogonal Sampling:
5. Pairwise Selection:
6. 1/µ Recombination Weights:
7. Sequential Selection:
8. Threshold Convergence:
9. Two-Point step-size Adaptation:
10. Increasing Population (IPOP/BIPOP):
11. Quasi-Gaussian Sampling (Sobol/Halton):

4,608 possible structures $\{0,1\}^9 \times \{0,1,2\}^2$

Meta-algorithm:

- (1,12)-GA
- self-adaptive mutation
- 240 evaluations (5.2% of search space)

Algorithm 1 Modular CMA-ES Framework

options $\leftarrow$ which modules are active
// Local restart loop
while not terminate **do**
  $t \leftarrow 0$
  $\bar{x} \leftarrow$ randomly generated individual
  SetParameters(init-params)
  // ES execution loop
  while not terminate local **do**
    params $\leftarrow$ Initialize(init-params)
    $\vec{x} \leftarrow$ Mutate($\bar{x}$, options) *// Sampler, Threshold*
    $\vec{f} \leftarrow$ Evaluate($\vec{x}$, options) *// Sequential*
    $P^{(t+1)} \leftarrow$ Select($\vec{x}$, $\vec{f}$, options) *// Elitism, Pairwise*
    $\bar{x} \leftarrow$ Recombine($P^{(t+1)}$, options) *// Weights*
    UpdateParams(params, options) *// Active, TPA*
    $t \leftarrow t + 1$
  end while
  AdaptParams(init-params) *// (B)IPOP*
end while

S. van Rijn, H. Wang, M. van Leeuwen, Th. Bäck: Evolving the Structure of Evolution Strategies. In: *IEEE Symposium Series on Computational Intelligence, Athens, Greece, Dec. 6-9, 2016.* IEEE Press, Piscataway, NJ, 2016.

Idea: ES-Superstructure

1. Active Update
2. Elitism
3. Mirrored Sampling
4. Orthogonal Sampling
5. Pairwise Sel
6. 1/
7.
8.

11.

4,608 possib

Meta-algorithm:
- (1,12)-GA
- self-adaptive mutation
- 240 evaluations (5.2% of search space)

Algorithm 1 Modular CMA-ES Framework

```
options ← which modules are active
// Local restart loop
while not terminate do
                                    idual
                                    old
                                    airwise
                                    hts
                                    // Active, TPA
  while
  AdaptParams(init-params)          // (B)IPOP
end while
```

4.608 Algorithms?

S. van Rijn, H. Wang, M. van Leeuwen, Th. Bäck: Evolving the Structure of Evolution Strategies. In: *IEEE Symposium Series on Computational Intelligence, Athens, Greece, Dec. 6-9, 2016.* IEEE Press, Piscataway, NJ, 2016.

Idea: ES-Superstructure

1. Active Update
2. Elitism
3. Mirrored Sampling
4. Orthogonal Sampling
5. Pairwise Sel
6. 1/
7.
8

11.

4,608 possib

Meta-algorithm:

- (1,12)-GA
- self-adaptive mutation
- 240 evaluations (5.2% of search space)

Algorithm 1 Modular CMA-ES Framework

options ← which modules are active
// Local restart loop
while not terminate **do**
idual
old
airwise
hts
// Active, TPA
AdaptParams(init-params) *// (B)IPOP*
end while

Combinatorial Algorithmics

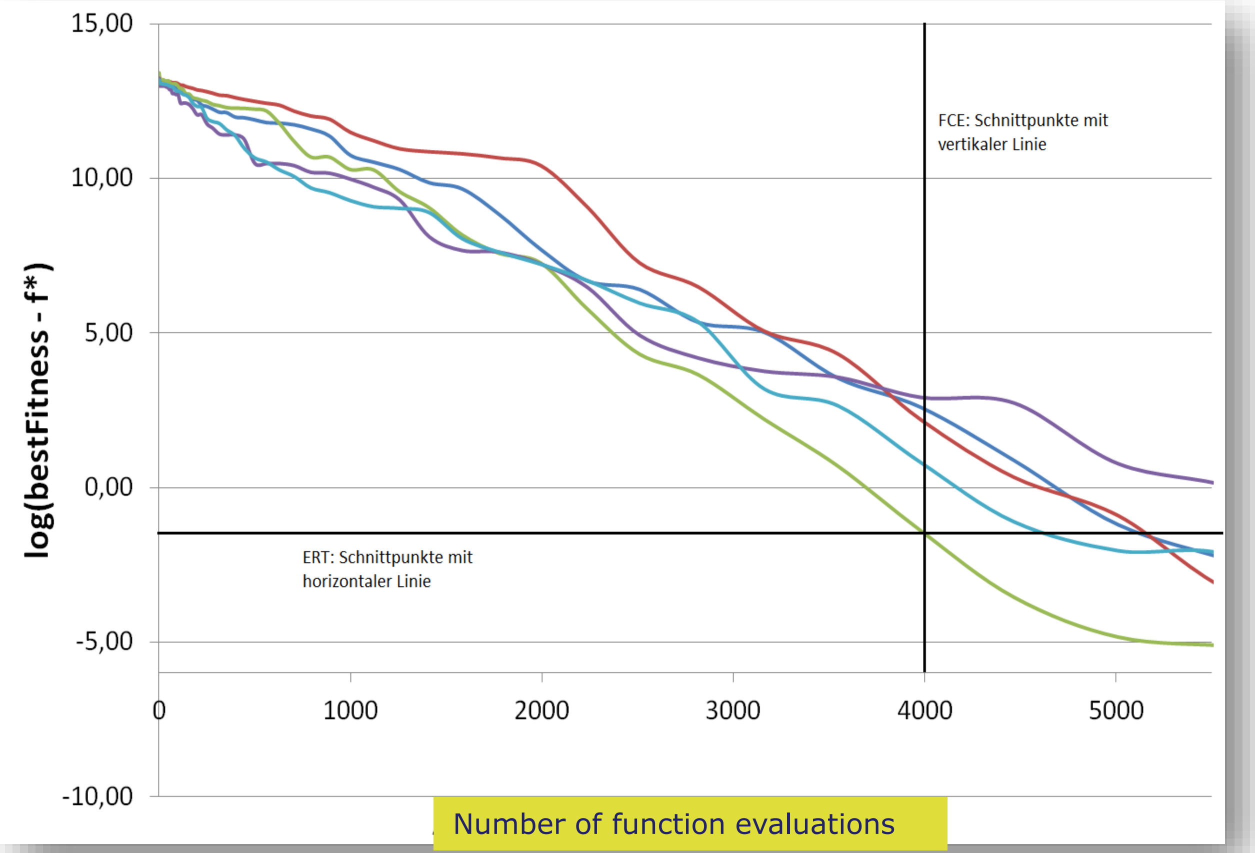
S. van Rijn, H. Wang, M. van Leeuwen, Th. Bäck: Evolving the Structure of Evolution Strategies. In: *IEEE Symposium Series on Computational Intelligence, Athens, Greece, Dec. 6-9, 2016.* IEEE Press, Piscataway, NJ, 2016.

Some Standard Algorithms

Strategy	Coding
CMA-ES	[0 0 0 0 0 0 0 0 0 0 0]
Active-CMA-ES	[1 0 0 0 0 0 0 0 0 0 0]
Elitist-CMA-ES	[0 1 0 0 0 0 0 0 0 0 0]
Mirrored-pairwise CMA-ES	[0 0 1 0 0 0 0 1 0 0 0]
IPOP-CMA-ES	[0 0 0 0 0 0 0 0 0 0 1]
Active-IPOP-CMA-ES	[1 0 0 0 0 0 0 0 0 0 1]
Elitist Active-IPOP-CMA-ES	[1 1 0 0 0 0 0 0 0 0 1]
BIPOP-CMA-ES	[0 0 0 0 0 0 0 0 0 0 2]
Active-BIPOP-CMA-ES	[1 0 0 0 0 0 0 0 0 0 2]
Elitist Active-BIPOP-CMA-ES	[1 1 0 0 0 0 0 0 0 0 2]

Efficieny Measures

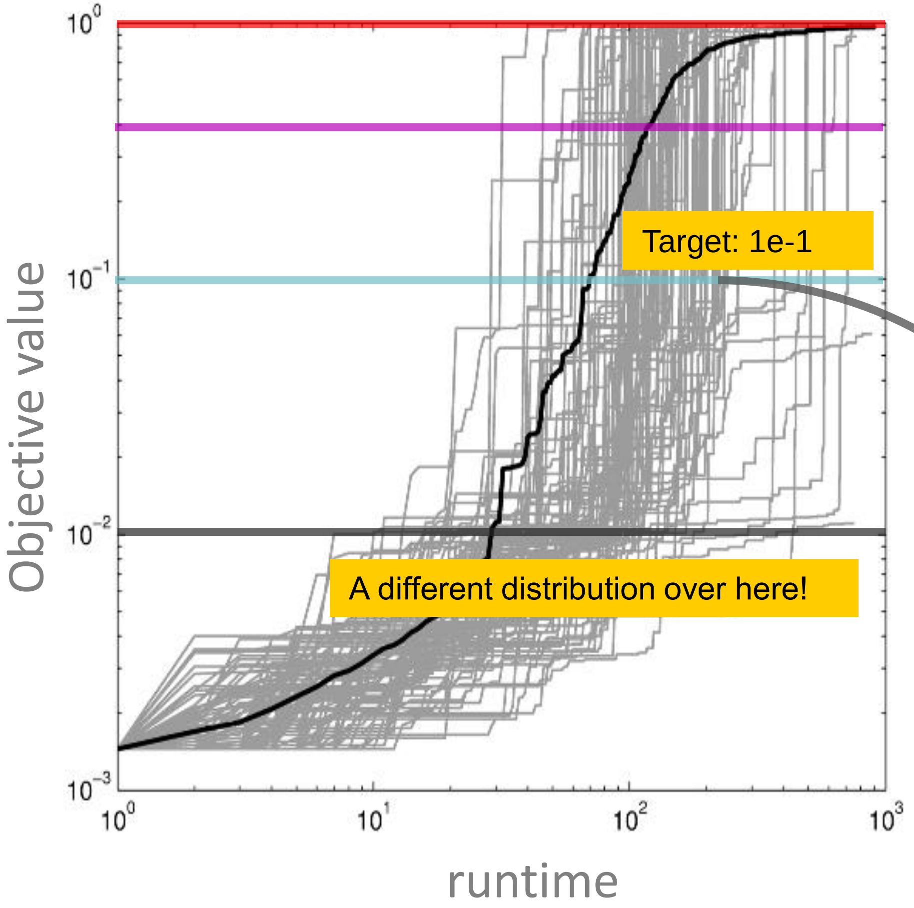
- Expected Running Time (ERT)
  - Horizontal viewpoint
  - Recommended in BBOB
  - Advantage due to quantitative comparison
  - For runs with very small number of function evaluations, choice of fixed error Δf* is problematic

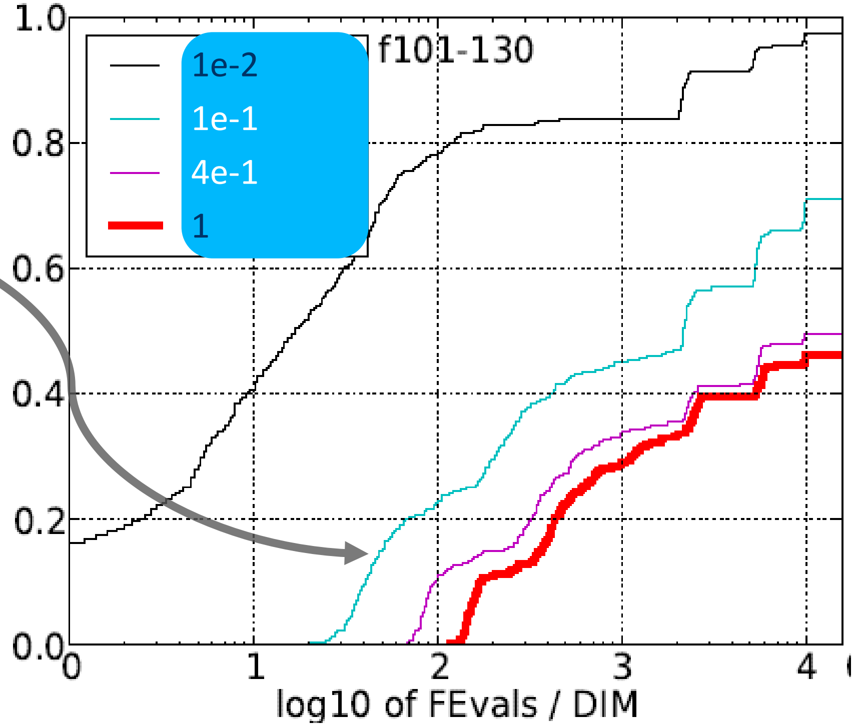


- Fixed Cost Error (FCE)
  - Vertical point of view
  - Allows only for qualitative comparison
  - Directly applicable for runs with very small number of function evaluations

Performance Measures

- Fixed-target and fixed-budget view
- ***Empirical Cumulative Distribution Functions*** (ECDFs)





Sample Results (F1, 20-D)

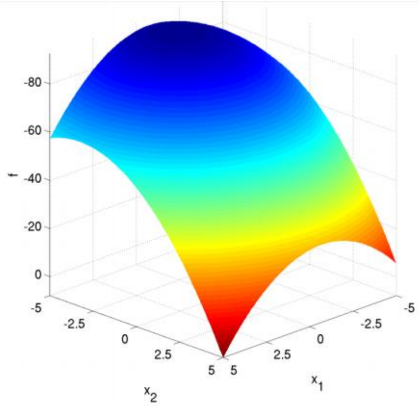
```
Results for F1 (Sphere) in 20dim:
_____________________________________________________________________________________
Default CMA-ES structures
CMA-ES                                ERT: 3430.875  (std: 1.67e+02)  Rank: 1656
Elitist CMA-ES                        ERT: 3121.125  (std: 1.85e+02)  Rank: 1389
Mirrored-pairwise CMA-ES              ERT: 3317.625  (std: 1.29e+02)  Rank: 1580
Elitist Active-IPOP-CMA-ES            ERT:  3050.25  (std: 2.34e+02)  Rank: 1238
Elitist Active-BIPOP-CMA-ES           ERT:   3132.0  (std: 2.15e+02)  Rank: 1399

Top-10 from Brute Force enumeration
[0 0 1 1 0 1 1 0 0 2 0]               ERT: 1285.375  (std: 1.07e+02)      Rank:    1
[0 0 1 1 0 1 1 0 0 2 2]               ERT: 1293.625  (std: 1.01e+02)      Rank:    2
[0 0 1 1 0 1 1 0 0 2 1]               ERT: 1328.125  (std:  1.1e+02)      Rank:    3
[0 0 1 1 0 1 1 1 0 2 2]               ERT: 1455.625  (std: 1.03e+02)      Rank:    4
[1 0 1 1 0 0 1 0 0 1 0]               ERT:   1483.0  (std:     77.2)      Rank:   10

Best as found 10 separate GA runs
[0 0 1 1 0 1 1 0 0 2 1]               ERT: 1328.125  (std:  1.1e+02)
[0 0 1 1 0 1 1 0 0 2 1]               ERT: 1328.125  (std:  1.1e+02)
[1 0 1 1 0 0 1 0 0 0 0]               ERT:  1467.25  (std:     95.7)
[0 0 1 1 0 1 1 0 0 2 2]               ERT: 1293.625  (std: 1.01e+02)
[1 0 1 1 0 0 1 0 0 1 1]               ERT:   1480.0  (std:     95.6)
[0 0 1 1 0 1 1 0 0 2 0]               ERT: 1285.375  (std: 1.07e+02)
[0 0 1 1 0 1 1 0 0 2 2]               ERT: 1293.625  (std: 1.01e+02)
[1 0 1 1 0 0 1 0 0 1 1]               ERT:   1480.0  (std:     95.6)
[1 0 1 1 0 0 1 0 0 0 0]               ERT:  1467.25  (std:     95.7)
[1 0 1 1 0 0 1 0 0 1 1]               ERT:   1480.0  (std:     95.6)
```

[1 1 0 0 0 0 0 0 0 0 1]

Mirrored + orthogonal sampling + 1/μ recombination weights + sequential selection + BIPOP



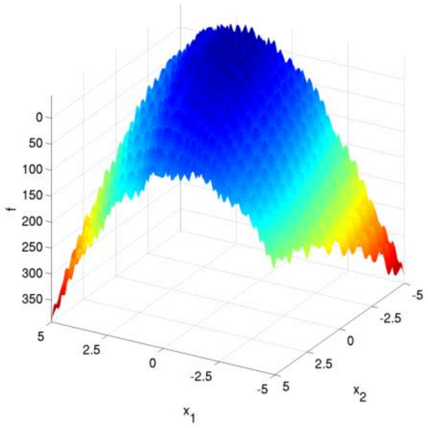
Sample Results (F15, 20-D)

```
Results for F15 (Multi-modal Rastrigin) in 20dim:
__________________________________________________________________________________
Default CMA-ES structures (NOTE: Active (B)IPOP is *without* elitism here)
CMA-ES                              FCE:     30.3   (std:     9.84)      Rank:  500
Elitist CMA-ES                      FCE:     63.0   (std:     34.5)      Rank: 1886
Mirrored-pairwise CMA-ES            FCE:     30.7   (std:     10.3)      Rank:  508
Active-IPOP-CMA-ES                  FCE:     27.2   (std:     12.7)      Rank:  390
Active-BIPOP-CMA-ES                 FCE:     22.9   (std:     9.45)      Rank:  258   [1 0 0 0 0 0 0 0 0 0 2]

Top-10 from Brute Force enumeration
[0 0 1 1 0 0 0 0 0 1 1]             FCE:     8.02   (std:     3.19)      Rank:    1
[0 0 1 1 0 0 0 0 0 0 1]             FCE:     8.46   (std:     2.91)      Rank:    2
[0 0 1 0 0 0 0 0 0 1 1]             FCE:     8.55   (std:     3.52)      Rank:    3
[0 0 0 0 1 0 0 0 0 1 1]             FCE:     9.11   (std:     3.18)      Rank:    4
[0 0 1 1 1 0 0 0 0 0 1]             FCE:     10.4   (std:     4.39)      Rank:   10

Best as found 10 separate GA runs
[0 0 1 1 0 0 0 0 0 1 1]             FCE:     8.02   (std:     3.19)
[0 0 1 1 0 0 0 0 0 1 1]             FCE:     8.02   (std:     3.19)
[0 0 1 1 0 0 0 0 0 0 1]             FCE:     8.46   (std:     2.91)
[0 0 1 1 0 0 0 0 0 1 1]             FCE:     8.02   (std:     3.19)
[0 0 1 1 0 0 0 0 0 1 1]             FCE:     8.02   (std:     3.19)
[0 0 1 1 0 0 0 0 0 1 1]             FCE:     8.02   (std:     3.19)
[0 0 1 1 0 0 0 0 0 1 1]             FCE:     8.02   (std:     3.19)
[0 0 1 1 0 0 0 0 0 1 1]             FCE:     8.02   (std:     3.19)
[0 0 1 1 0 0 0 0 0 1 1]             FCE:     8.02   (std:     3.19)
[0 0 1 1 0 0 0 0 0 1 1]             FCE:     8.02   (std:     3.19)
```

Mirrored + orthogonal sampling + IPOP + Sobol sampling



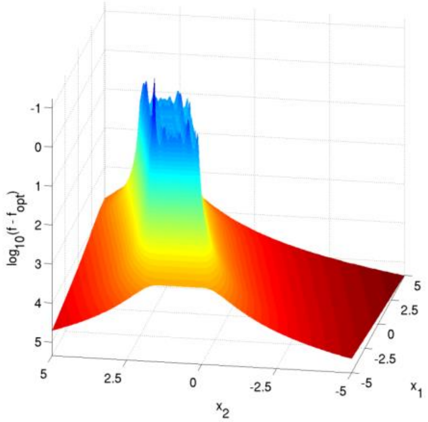
Sample Results (F20, 20-D)

Results for F20 (Schwefel) in 20dim:

Default CMA-ES structures					
CMA-ES	FCE:	1.89	(std: 0.226)	Rank: 2382	
Elitist CMA-ES	FCE:	1.79	(std: 0.25)	Rank: 1411	
Mirrored-pairwise CMA-ES	FCE:	1.77	(std: 0.23)	Rank: 1109	[0 0 1 0 0 0 0 1 0 0 0]
Elitist Active-IPOP-CMA-ES	FCE:	1.79	(std: 0.211)	Rank: 1364	
Elitist Active-BIPOP-CMA-ES	FCE:	1.77	(std: 0.189)	Rank: 1128	

Top-10 from Brute Force enumeration				
[1 1 0 1 1 0 0 1 0 2 2]	FCE:	1.52	(std: 0.241)	Rank: 1
[1 0 0 1 1 0 0 1 0 2 1]	FCE:	1.52	(std: 0.195)	Rank: 2
[0 0 1 0 1 0 0 1 0 2 0]	FCE:	1.54	(std: 0.29)	Rank: 3
[0 0 0 0 1 0 0 1 0 2 1]	FCE:	1.54	(std: 0.217)	Rank: 4
[0 1 1 1 0 0 0 1 0 1 2]	FCE:	1.56	(std: 0.221)	Rank: 10

Best as found 10 separate GA runs			
[1 1 0 1 1 0 0 1 0 2 2]	FCE:	1.52	(std: 0.241)
[1 0 0 1 1 0 0 1 0 2 1]	FCE:	1.52	(std: 0.195)
[1 1 0 1 1 0 0 1 0 2 2]	FCE:	1.52	(std: 0.241)
[0 1 0 1 0 0 1 1 0 0 2]	FCE:	1.55	(std: 0.306)
[0 0 1 0 0 0 0 1 0 0 2]	FCE:	1.55	(std: 0.168)
[1 1 0 1 1 0 0 1 0 2 2]	FCE:	1.52	(std: 0.241)
[0 0 1 0 0 0 1 1 0 2 0]	FCE:	1.57	(std: 0.246)
[0 1 0 0 1 0 0 1 1 1 2]	FCE:	1.6	(std: 0.157)
[1 1 0 1 1 0 0 1 0 2 2]	FCE:	1.52	(std: 0.241)
[0 1 1 1 0 0 0 1 0 1 2]	FCE:	1.56	(std: 0.221)



Active update + elitism + mirrored + orthogonal sampling + threshold convergence + BIPOP + Halton sampling

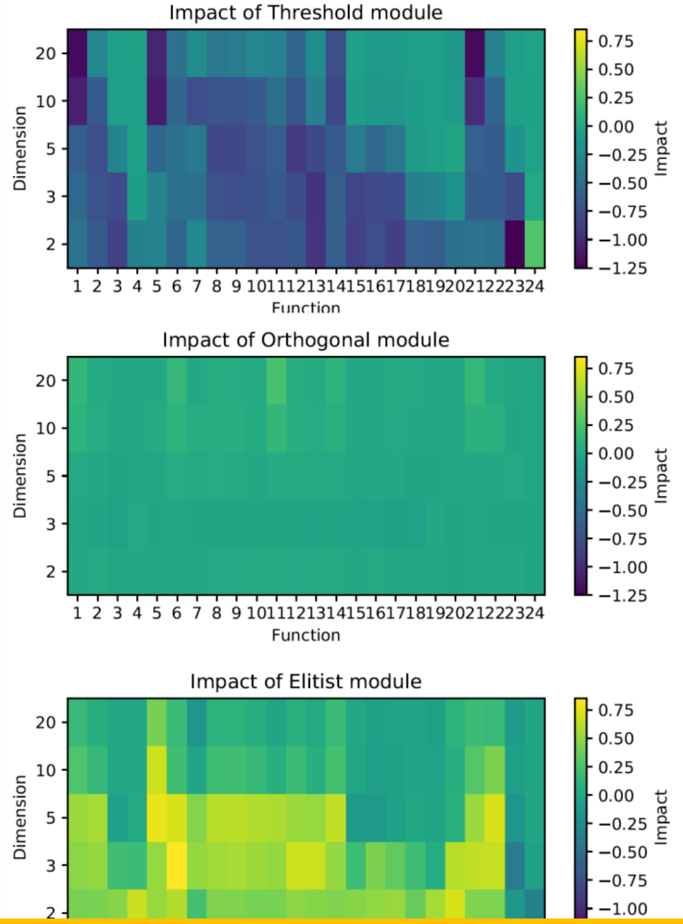
Mining the Results

Impact scores (all test functions)

Table 3: ***Impact score and statistical significance per module.*** **The values shown are calculated using aggregated performance data on all** 4, 608 **configurations for each of the** 120 **experiments. P-values are calculated using the two-tailed *Mann-Whitney U* test.**

Module name	I_{module}	p-value
Active	-0.13	0.0
Elitism	**0.27**	**0.0**
Mirrored	0.01	$6.61 \cdot 10^{-10}$
Orthogonal	0.02	$8.72 \cdot 10^{-43}$
Sequential Selection	-0.10	0.0
Threshold	**-0.49**	**0.0**
TPA	0.07	0.0
Pairwise Selection	-0.03	$3.13 \cdot 10^{-43}$
Weights	-0.12	0.0
Base-Sampler	0.02	$1.18 \cdot 10^{-15}$
(B)IPOP	0.10	0.0

Impact by function and dimension



Module interaction impact

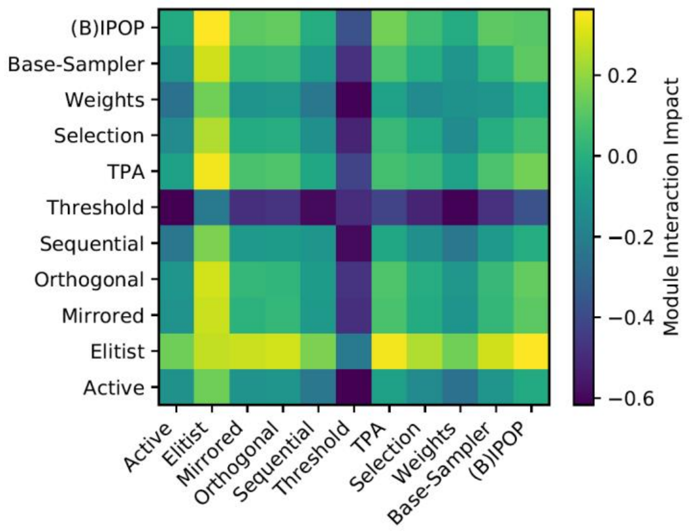


Figure 3: *Aggregate Module Interaction Impact.* Displayed impact scores are calculated as a mean over all 120 experiments. Each score indicates the difference in q-score between the set of configurations with two modules x and y active, versus the remaining configurations. The heatmap is symmetrical along the diagonal $x = y$, which indicates the single module impact as shown in Table 3.

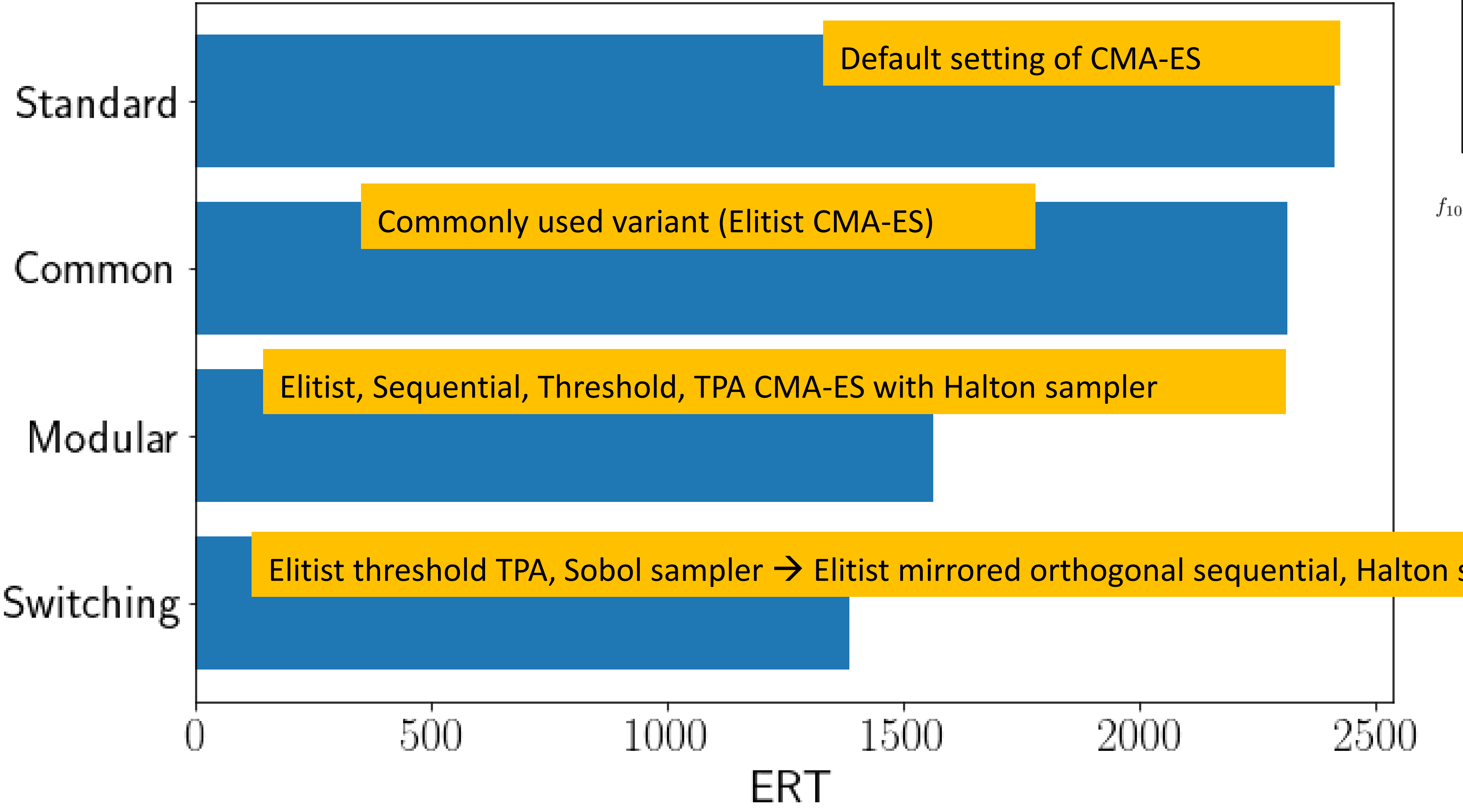
S. van Rijn, H. Wang, B. van Stein, T. Bäck: *Algorithm Configuration Data Mining for CMA Evolution Strategies.* Proceedings of the Genetic and Evolutionary Computation Conference 2017, Berlin, Germany, July 15-19, 2017 (GECCO´17).

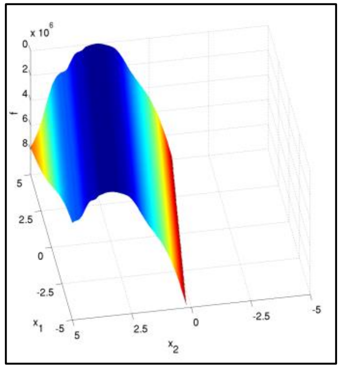
3.3

ONLINE SELECTION OF VARIANTS

D. Vermetten, S. van Rijn, T. Bäck, C. Doerr: *Online Selection of CMA-ES Variants.* Proceedings of the Genetic and Evolutionary Computation Conference 2019, Prague, Czech Republic, July 13-17, 2019 (GECCO ´19).

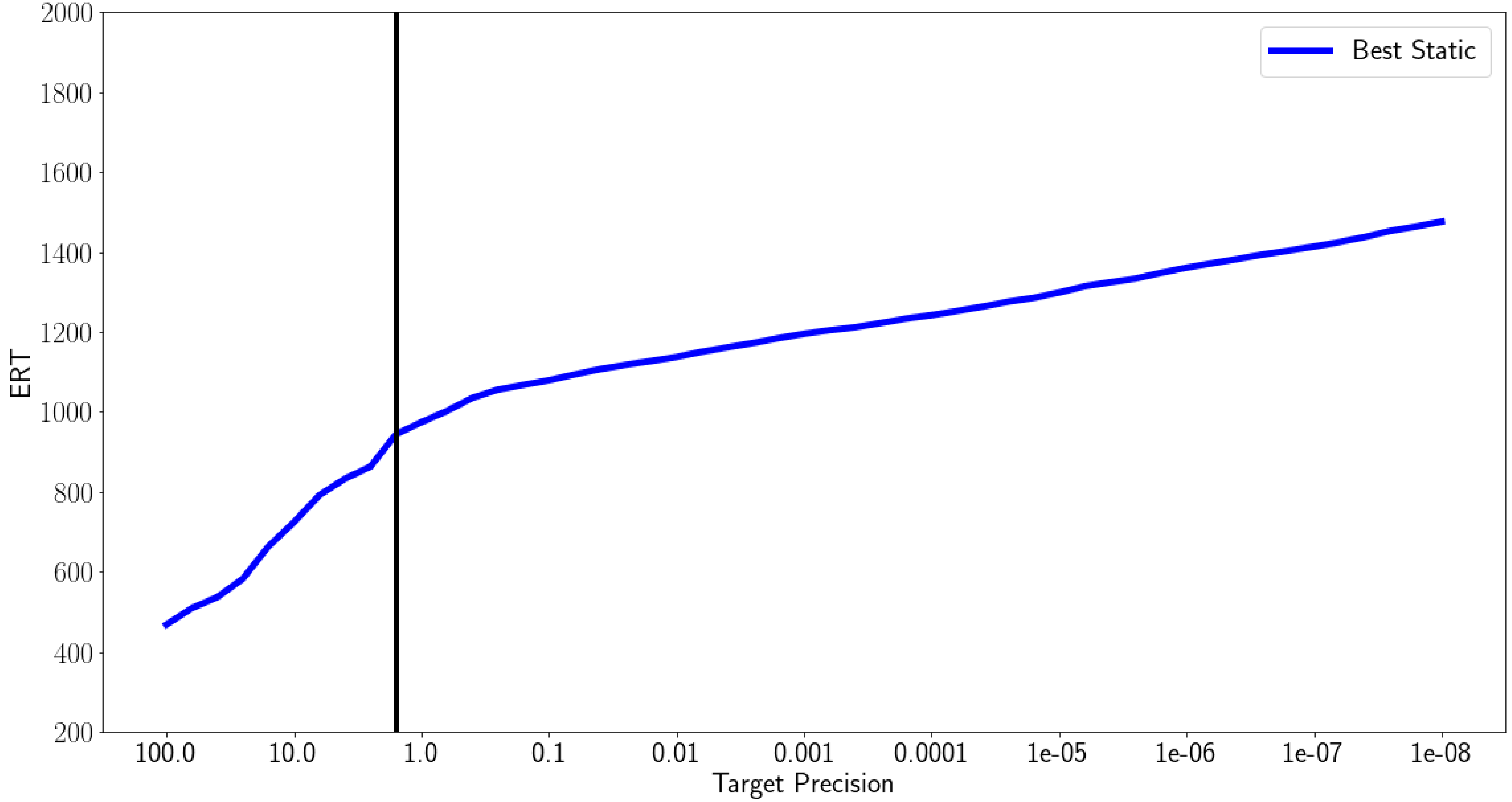
Results (F10)



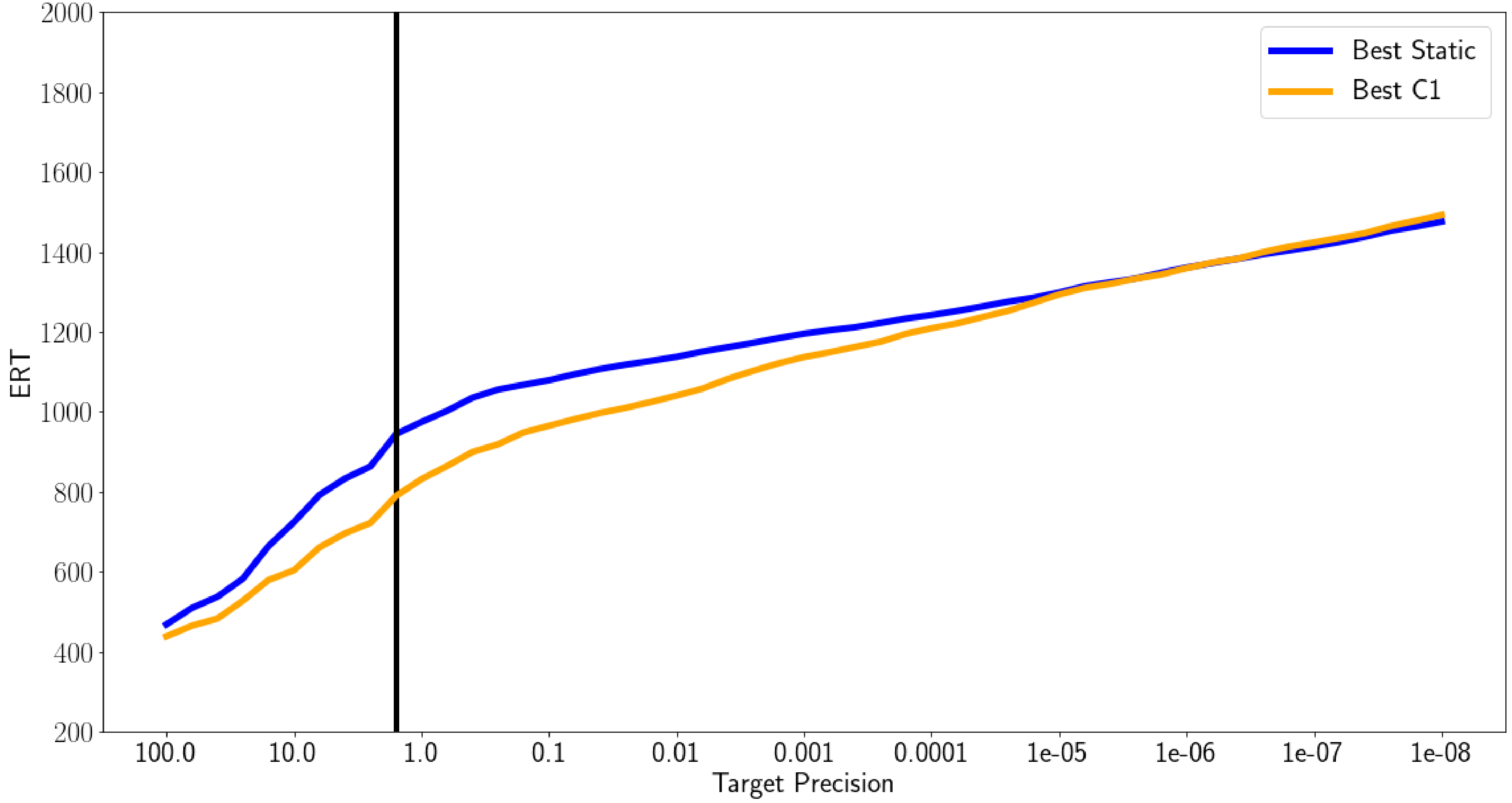


$$f_{10}(\mathbf{x}) = \sum_{i=1}^{D} 10^{6\frac{i-1}{D-1}} z_i^2 + f_{\mathrm{opt}}$$

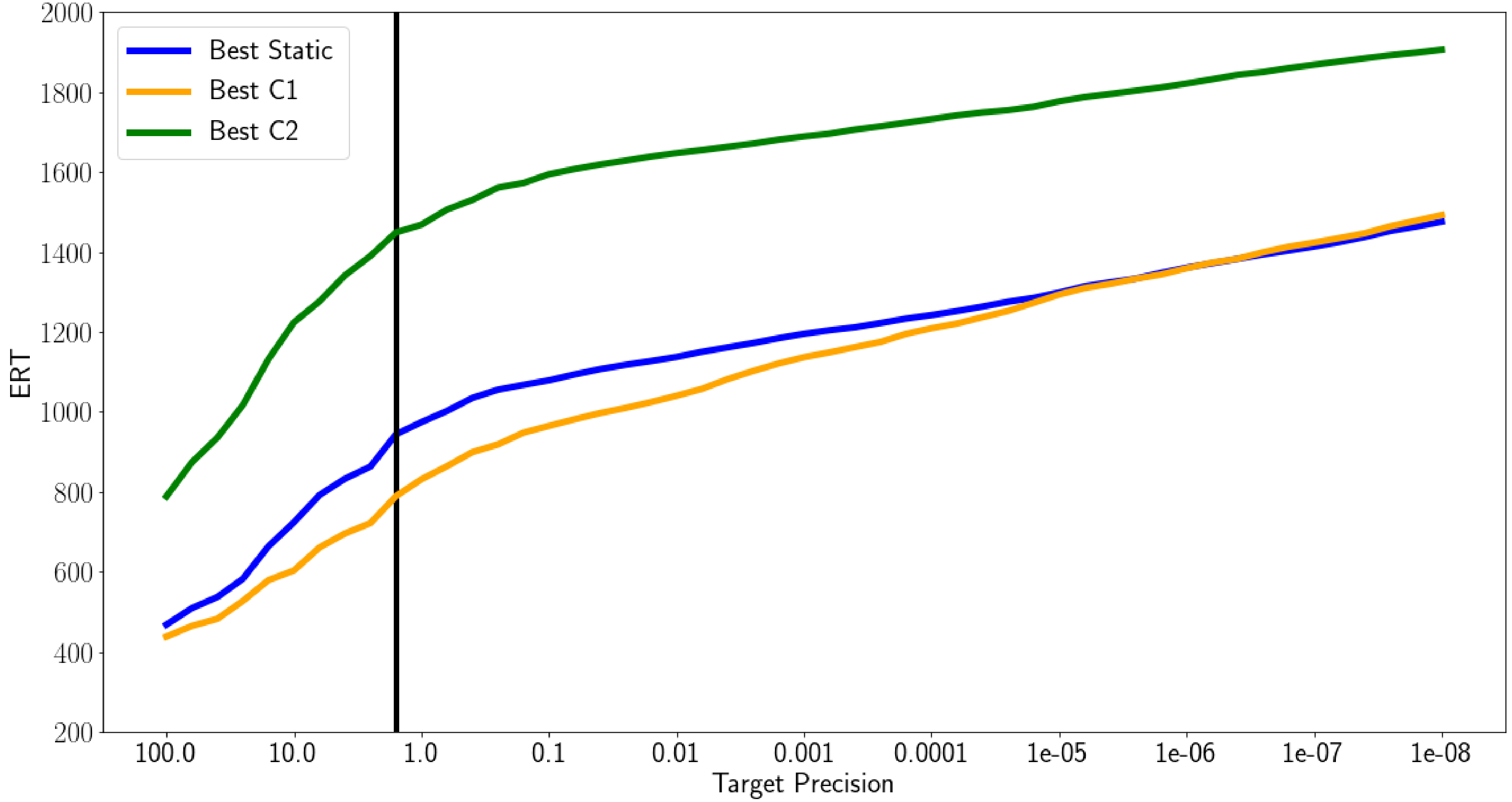
Idea Behind Configuration Switching



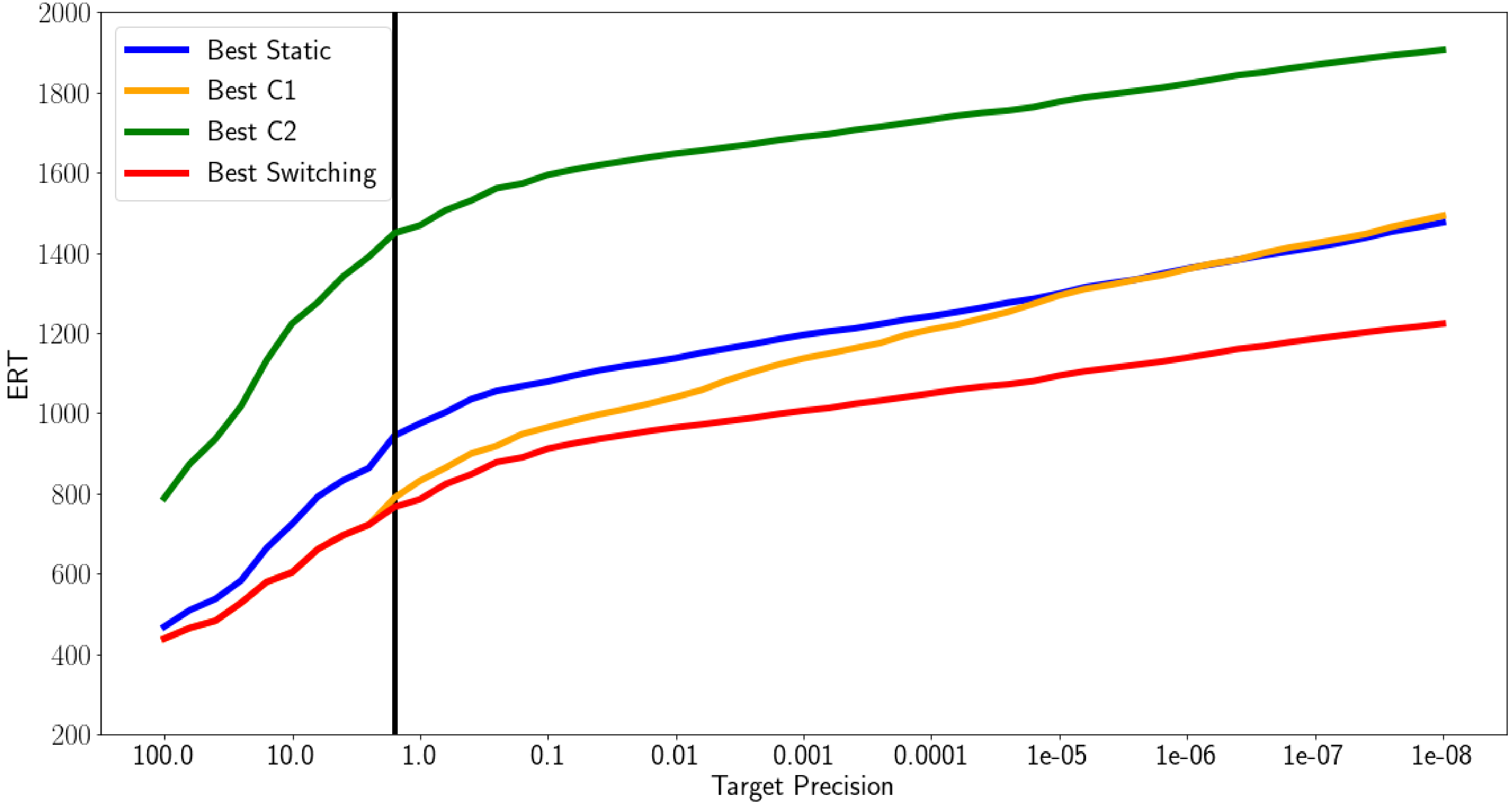
Idea Behind Configuration Switching



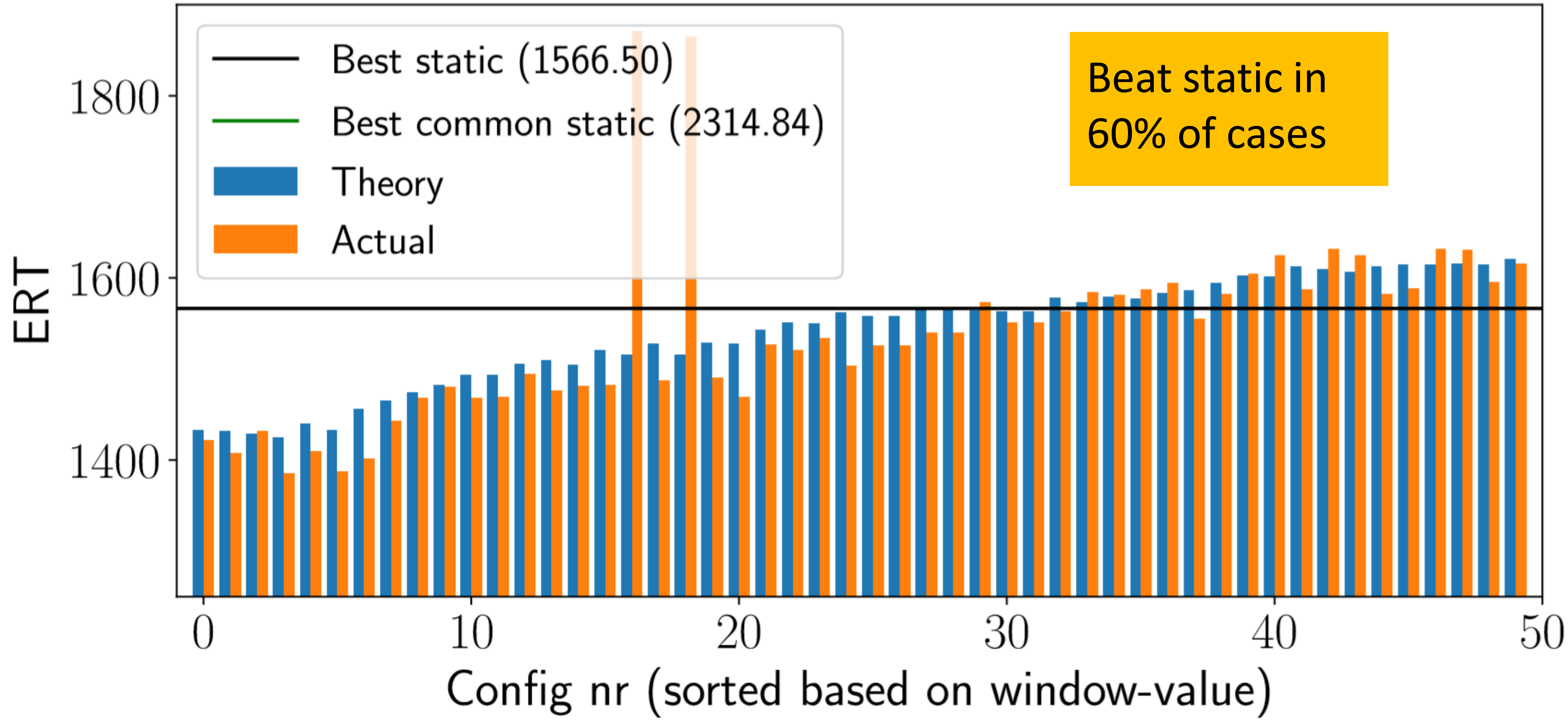
Idea Behind Configuration Switching



Idea Behind Configuration Switching



Zooming in on F10



Beat static in 60% of cases

Zooming in on F10

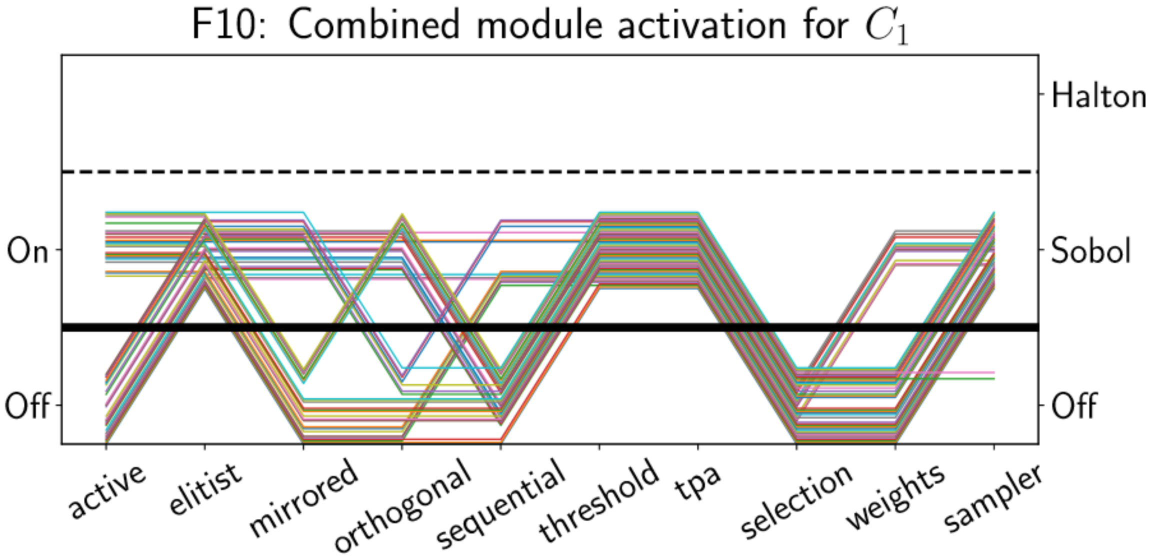
- Visually obvious difference between C_1 and C_2

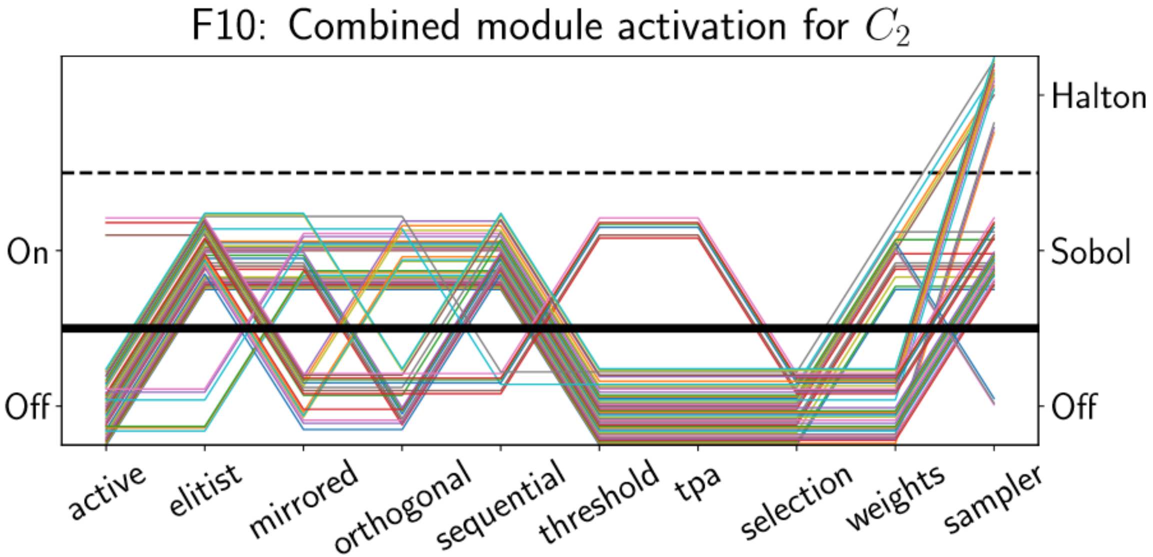
- Define Maximum module difference:

$$\frac{100}{n} \max_{Module\ m} \left| \sum_{1}^{n} m(C_1(i)) - \sum_{1}^{n} m(C_2(i)) \right|$$

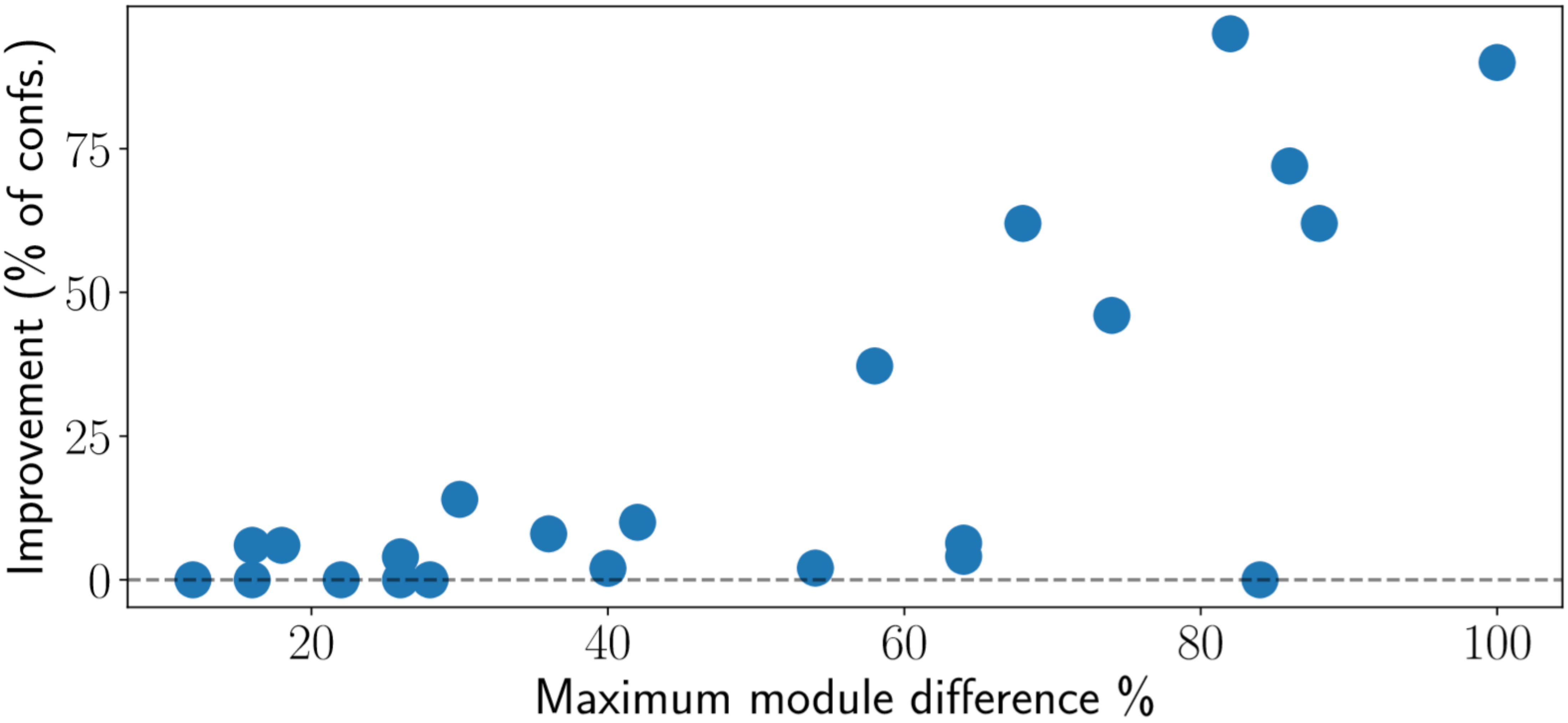
- Here: 88%

- Selection based on behavior of configuration





General Trend



4

CONCLUDING

Concluding Remarks

- Evolutionary Algorithms are powerful general-purpose optimizers
- Can be used for algorithm configuration as well
- Combinatorial algrithmics as an idea for generating new algorithms
- Next steps:
  - Online learning
  - Automatically generating the optimal optimization algorithm

Current Projects

- Process Mining by Multi-Objective Online Control
  - NWO TA
- Data Mining on High-Volume Simulation Output
  - NWO TA
- A Systems Approach towards Data Mining and Prediction in Airline Operations
  - NWO Indo-Dutch Joint Research Progr. for ICT
- Cross-Industry Predictive Maintenance Optimization Platform
  - STW Smart Industry
- Experience-Based Computation: Learning to Optimise
  - EU Marie Curie Industrial Training Network

Thomas Bäck (Natural Computing Group)

Optimization:

- Evolutionary Computation
- Optimization of Optimizers
- Optimizer Configuration
- Efficient Global Optimization
- Bayesian Global Optimization
- Quantum Computing for Optimization
- (Multi-Objective Optimization)

Data Science:

- Supervised Learning
- Gaussian Processes
- (Fuzzy) Cluster Kriging
- Imputation Methods
- Automatic data-driven modeling with RF, SVM, decision trees …

Machine Learning:

- Algorithm Selection and Configuration
- Automatically optimizing deep networks
- Learning rules from data
- Anomaly detection

My interest:

Making Optimization Algorithms Learn

Automatic Optimizer Optimization

Demonstrating the Value in Applications

Integrating Optimization and Machine Learning in many ways

Volkswagen

Europe
Honda Research Institute

GE Aviation

TATA STEEL

AIRFRANCE INDUSTRIES KLM Engineering & Maintenance

D-Wave The Quantum Computing Company™

QUALOGY

DAF

GOMECSYS fuel saving engine technology

Thomas Bäck (Natural Computing Group)

Optimization:

- Evolutionary Computation
- Optimization of Optimizers
- Optimizer Configuration
- Efficient Glob
- Baye
-

Data Science:

- Supervised Learning

Machine Learning:

- Algorithm Selection and

From Data and Models to Optimal Decisions

Automatic Optimizer Optimization

Demonstrating the Value in Applications

Integrating Optimization and Machine Learning in many ways

Development Team

Hao Wang h.wang@liacs.leidenuniv.nl
Leiden Institute of Advanced Computer Science

Diederick Vermetten d.l.vermetten@liacs.leidenuniv.nl
Leiden Institute of Advanced Computer Science

Furong Ye f.ye@liacs.leidenuniv.nl
Leiden Institute of Advanced Computer Science

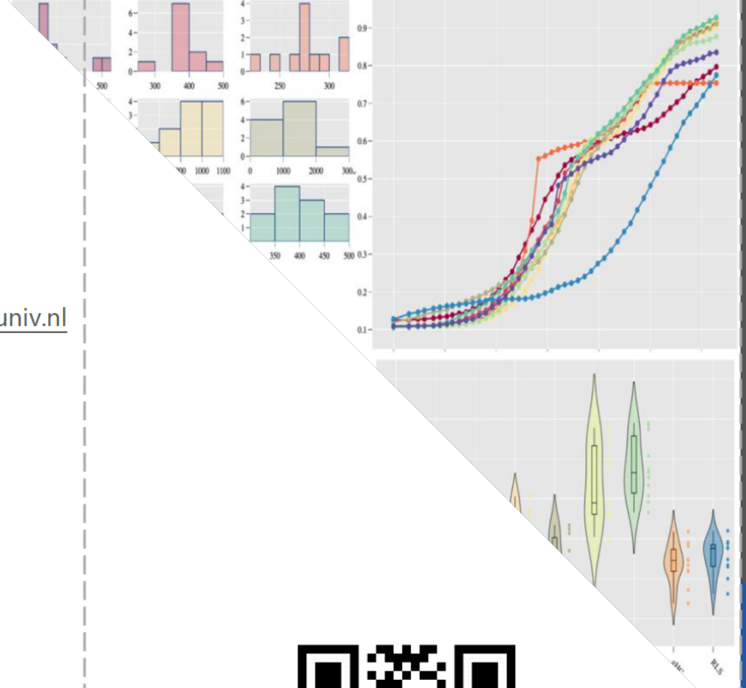
Carola Doerr Carola.Doerr@mpi-inf.mpg.de
CNRS and Sorbonne University

Ofer M. Shir ofersh@telhai.ac.il
Tel-Hai College

Thomas Bäck t.h.w.baeck@liacs.leidenuniv.nl
Leiden Institute of Advanced Computer Science

Acknowledgments

This work was supported by the Chinese scholarship council (CSC No. 201706310143), a public grant as part of the Investissement d’ avenir project, reference ANR-11-LABX-0056-LMH, LabEx LMH, in a joint call with the Gaspard Monge Program for optimization, operations research, and their interactions with data sciences, by Paris Ile-de-France Region, and by COST Action CA15140 “Improving Applicability of Nature-Inspired Optimisation by Joining Theory and Practice (ImAppNIO)” .



Contact us

Source code:
https://github.com/IOHprofiler

Web-based version:
http://iohprofiler.liacs.nl

Email:
iohprofiler@liacs.leidenuniv.nl

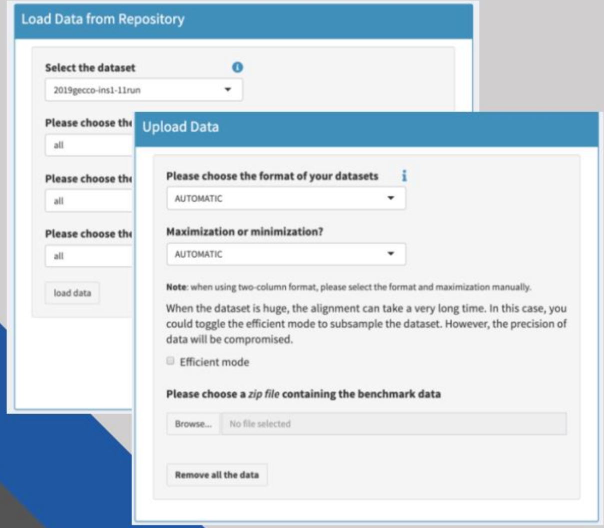
Documentation:
https://arxiv.org/abs/1810.05281

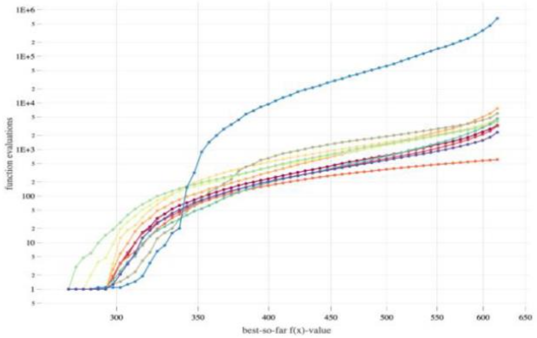
IOHprofiler

Leiden University
Tel-Hai College
CNRS
Sorbonne University

IOHprofiler is a novel tool for analyzing and comparing iterative optimization heuristics (IOHs), such as genetic algorithms, evolution strategies, local search algorithms, estimation of distribution algorithms, swarm optimization algorithms, etc. by providing detailed performance statistics.

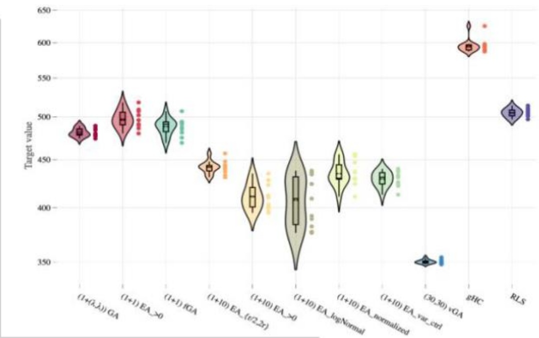
IOHprofiler also allows to track the evolution of internal states of IOHs, e.g., current solution, function value, and algorithm parameters, making it particularly useful for the analysis, comparison, and design of optimization algorithms. This tool is implemented as two software packages: **IOHexperimenter** and **IOHanalyzer**.

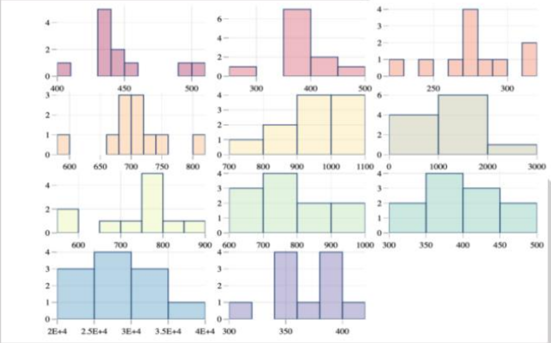




	algId	target	mean	median	sd	2%	5%	10%	25%	50%	75%	90%	95%	98%
1	(1+(λ,λ)) GA	600	2378	2360	209.07	1988	1988	1988	2250	2360	2461	2651	2651	269
2	(1+1) EA_>0	600	2114.27	2116	213.32	1797	1797	1797	1907	2116	2275	2330	2330	245
3	gHC	600	575.45	574	6.65	566	566	566	569	574	579	583	583	58
4	(1+10) EA_(r/2,2r)	600	4964.91	4956	442.67	4086	4086	4086	4714	4956	5084	5362	5362	583
5	(1+10) EA_>0	600	3391.55	3396	155.01	3076	3076	3076	3266	3396	3483	3557	3557	358
6	(1+10) EA_logNormal	600	4251.82	4093	643.14	3489	3489	3489	3804	4093	4464	4998	4998	553
7	(1+10) EA_normalized	600	3162.18	3173	179.59	2827	2827	2827	3036	3173	3310	3369	3369	337
8	(1+10) EA_var_ctrl	600	3080.91	3096	131.21	2913	2913	2913	2963	3096	3141	3221	3221	334
9	(1+1) fGA	600	2933.18	2935	306.38	2435	2435	2435	2728	2935	3036	3078	3078	365
10	(30,30) vGA	600	358746.91	347996	31706.41	322774	322774	322774	332515	347996	380811	405370	405370	40903

Showing 1 to 10 of 11 entries Previous 1 2 Next





IOHanalyzer is the data analysis and visualization module. A web-based version is hosted at http://iohprofiler.liacs.nl. It takes the data set generated by IOHexperimenter or COCO[1] and generates statistics for fixed-target running time / fixed-budget function value (mean, quantiles, etc.). ECDF curves are also available. More statistical procedures will be added.

IOHexperimenter provides an extensible experiment environment for generating performance data that can be interpreted by **IOHanalyzer**. It allows for testing your own algorithm on your own benchmark problems, or comparing to available data from the repository. A data repository is maintained at https://github.com/IOHprofiler/IOHdata, currently containing results from 11 algorithms on 23 functions and 4 dimensions.

[1] https://github.com/numbbo/coco

PPSN 2020

- Sep. 5-9, 2020, in Leiden, NL!
  - Easy to access
  - Organizers:
    - M. Preuss, T. Bäck, M. Emmerich, H. Trautmann
    - C. Doerr, O. Shir, H. Wang, A. Plaat
  - Honorary Chairs:
    - H.-P. Schwefel
    - G. Rozenberg

The Natural Computing Group