

Automatic Configuration and Learning for Evolutionary Computation

SSBSE 2019, 31.08.2019



Thomas Bäck, Leiden Institute of Advanced Computer Science



Universiteit
Leiden
The Netherlands

Overview

1.1 – Introduction to my World

1.2 – Application Examples

2.1 – Self-Adapting Optimization Algorithms

2.2 – State of the art: CMA-ES

3.1 – Optimization for Algorithm Configuration

3.2 – Algorithm Configuration for CMA-ES

3.3 – Online Selection of Variants

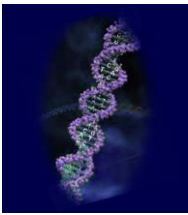
4 – Conclusions

1.1

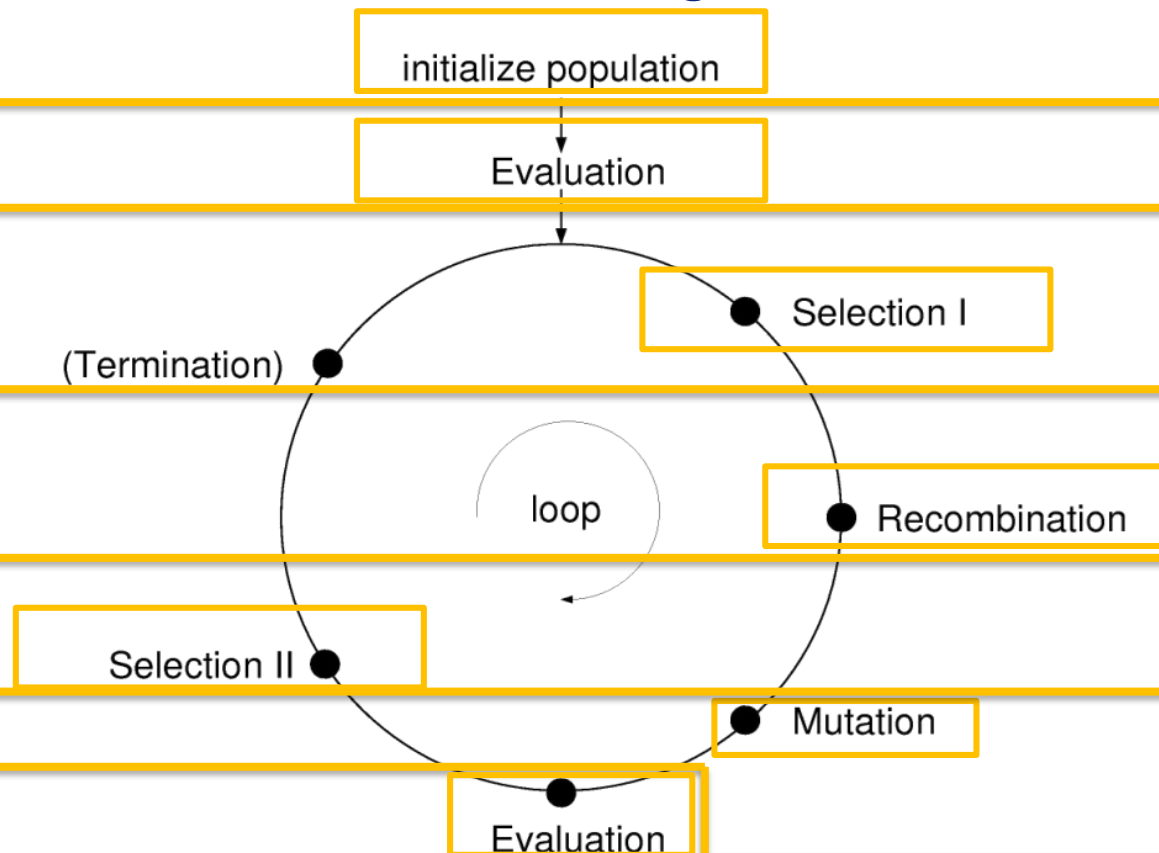
INTRODUCTION TO MY WORLD

Evolutionary Computation

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



- Global, direct search algorithms



Genetic Algorithm

$$\vec{a} \in \{0,1\}^n$$

Random

Objective function:

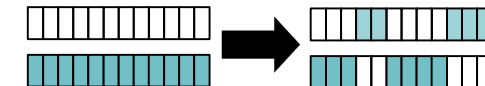
$$f(\vec{a}) \rightarrow \max$$

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

$$X_i \sim N(0, \sigma^2) \quad X \sim N(\mathbf{0}, \Sigma)$$

Objective function:

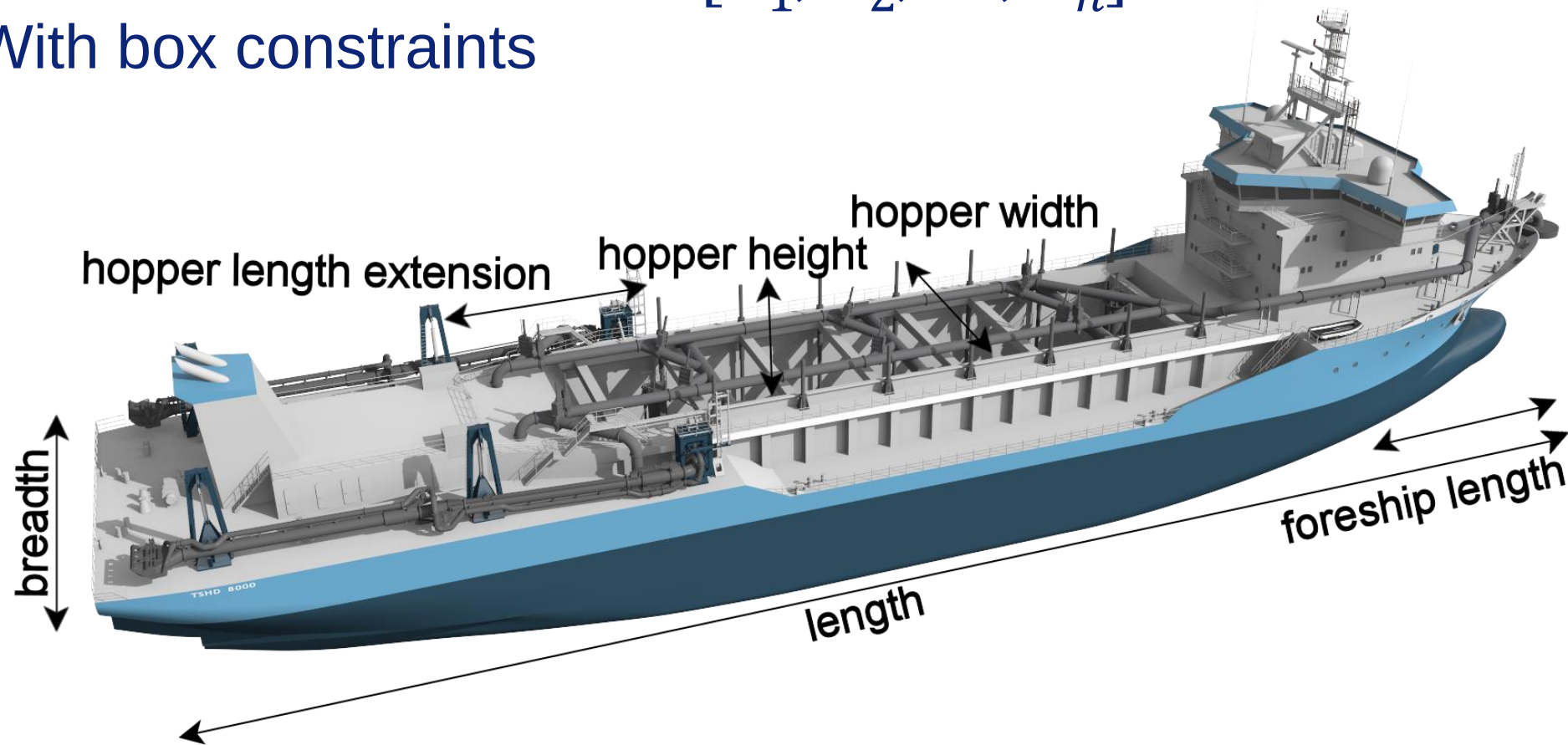
$$f(\vec{a}) \rightarrow \max$$

1.2

APPLICATION EXAMPLES

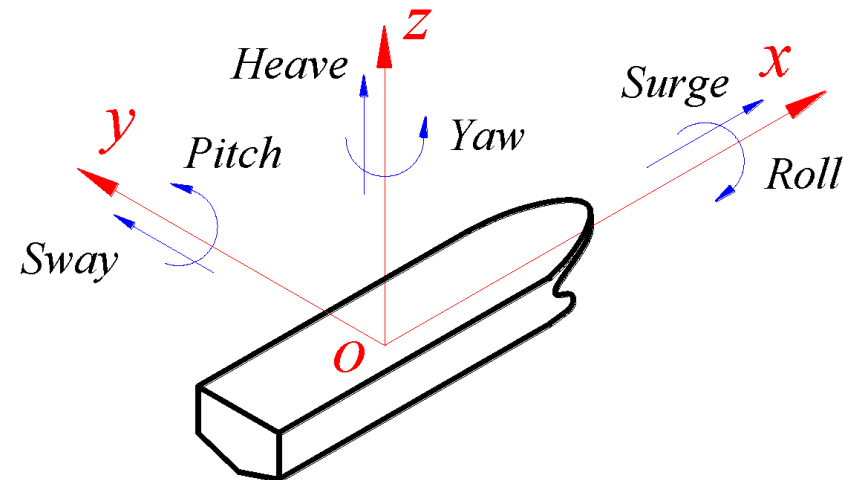
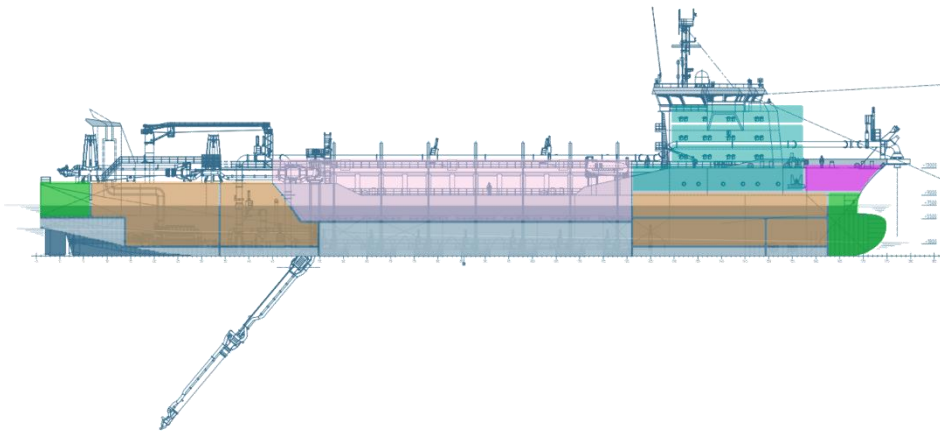
Multiojective Ship Design: Dredger

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



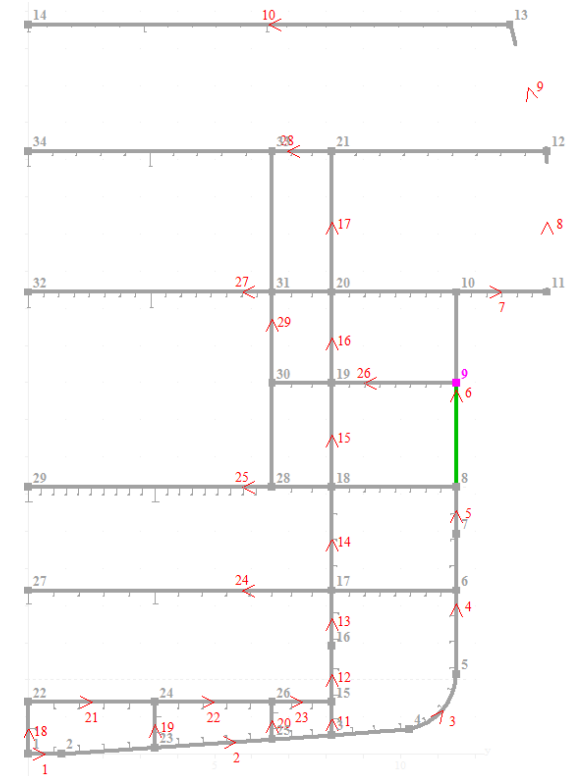
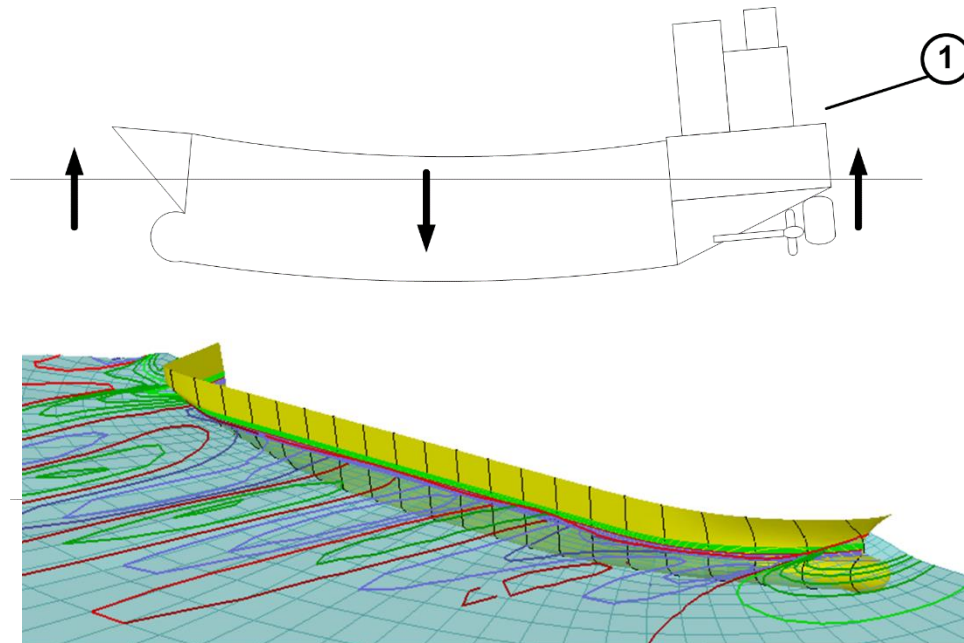
Dredger ship case

- Constraints : $G_j(\vec{X}) \leq 0 \quad \forall_j \in \{1, \dots, m\}$
 - Space reservation: payload, fuel tank, engine, dredging pump, accommodation
 - Regulating authorities: stability, strength, trim, heel



Dredger ship case

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

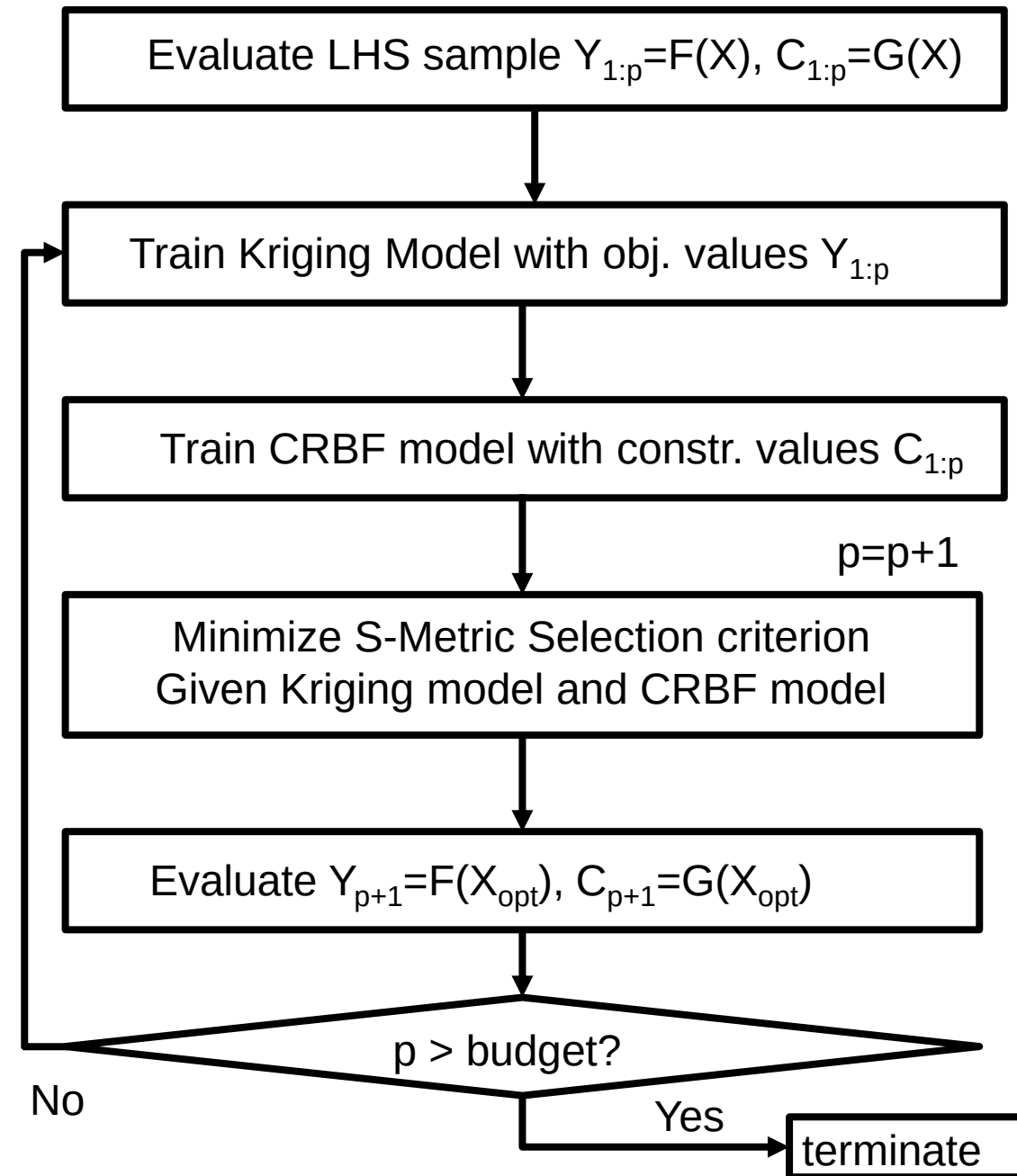


CEGO

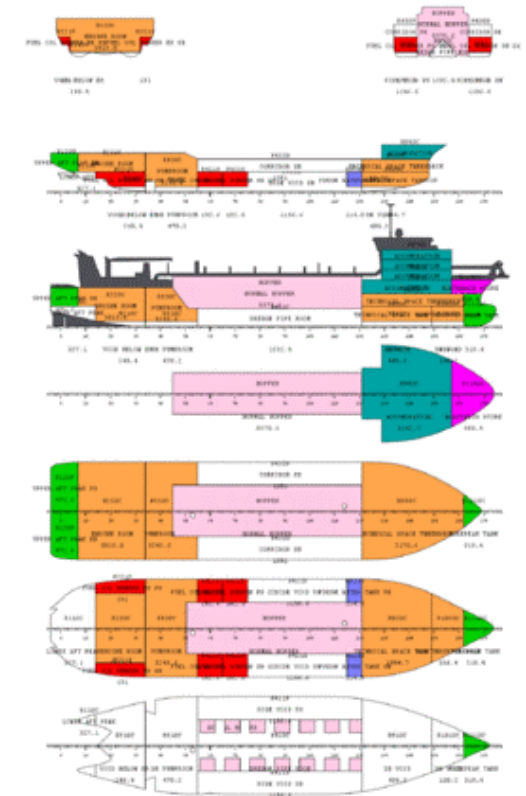
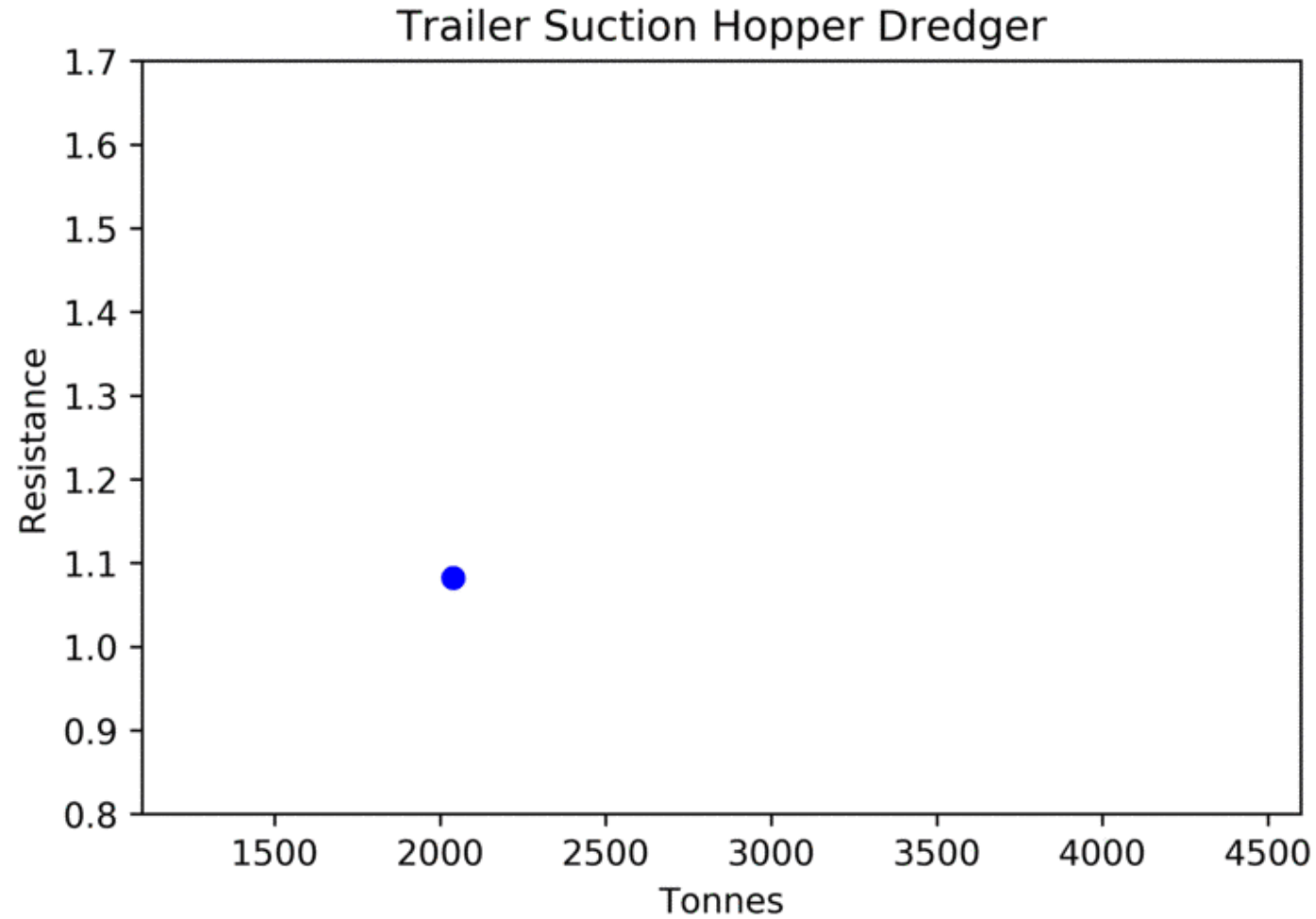
R. DE WINTER, B. VAN STEIN, M. DIJKMAN, AND T. BÄCK (2018). Designing Ships using Constrained Multi-Point Efficient Global Optimization. *Machine Learning, Optimization, and Data Science Conference*, 2018.

- 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.

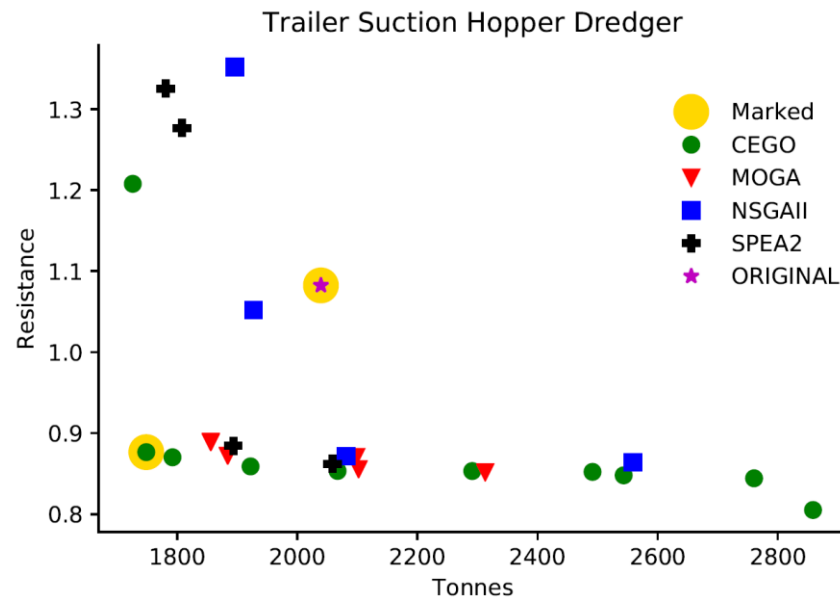


CEGO 200 iterations Dredger

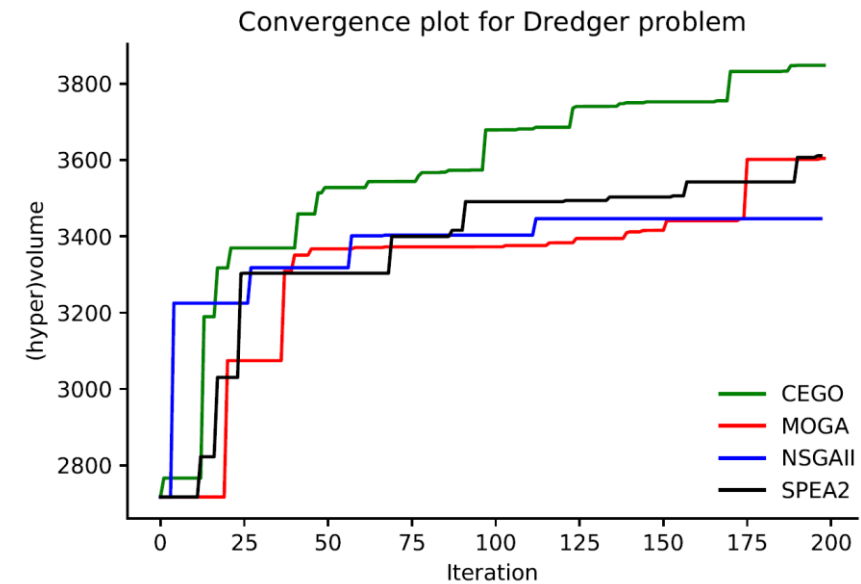


Algorithm Comparison

Pareto Frontier



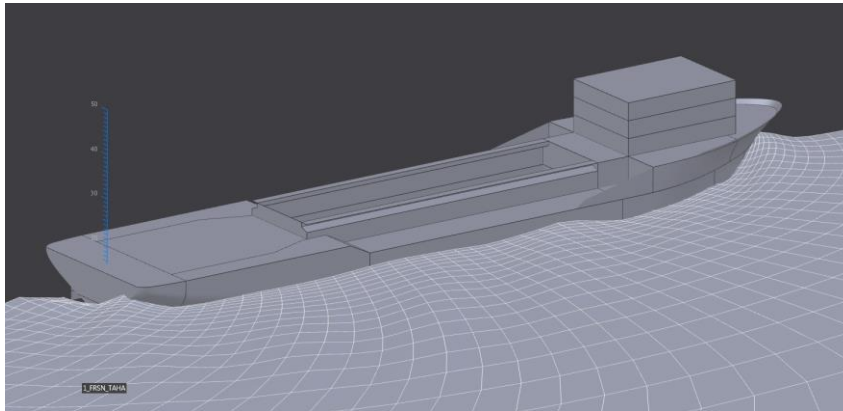
Convergence Plot



Marked individuals

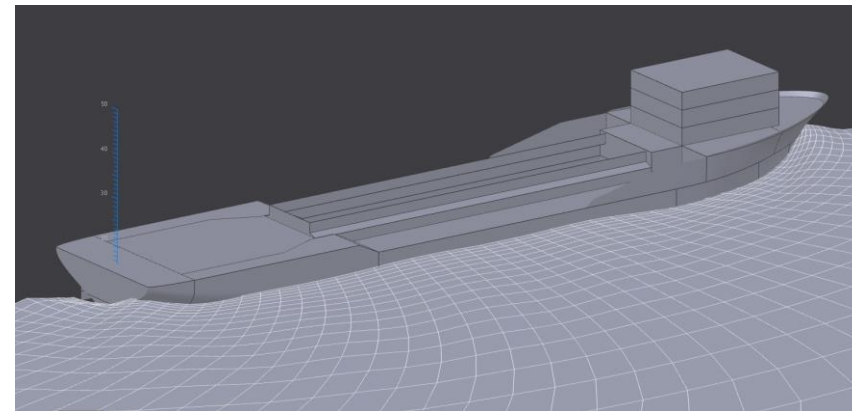
Original design

- Resistance coef: 1.08
- Steel weight: 2039Tonnes

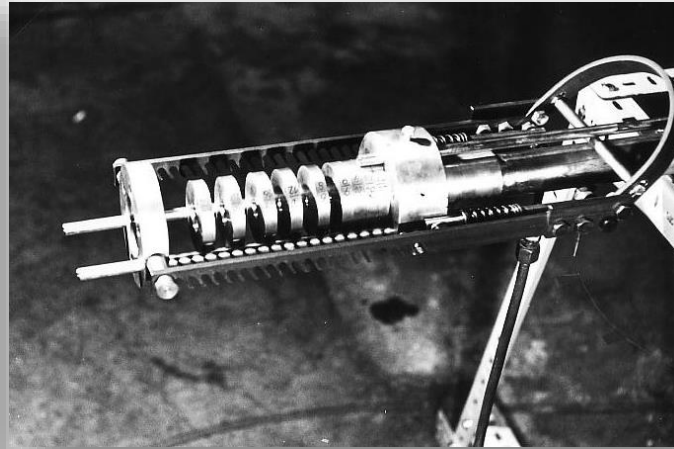


Marked CEGO result

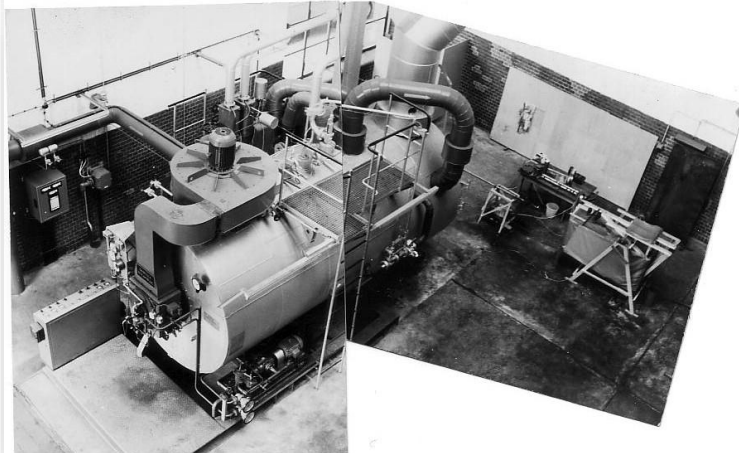
- Resistance coef: 0.877
- Steel weight: 1748Tonnes



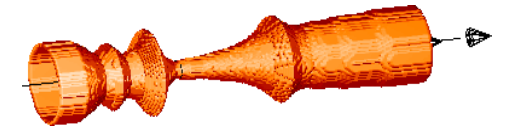
What if ... You Had Very Few Trials?



All photos
courtesy of
Hans-Paul
Schwefel.



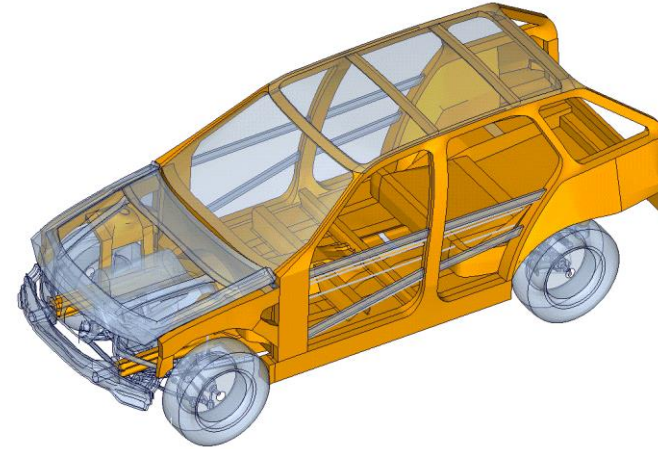
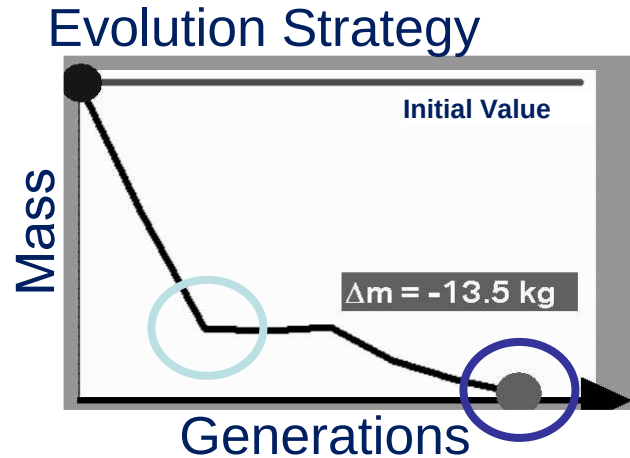
255 experiments



32% efficiency
improvement!

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

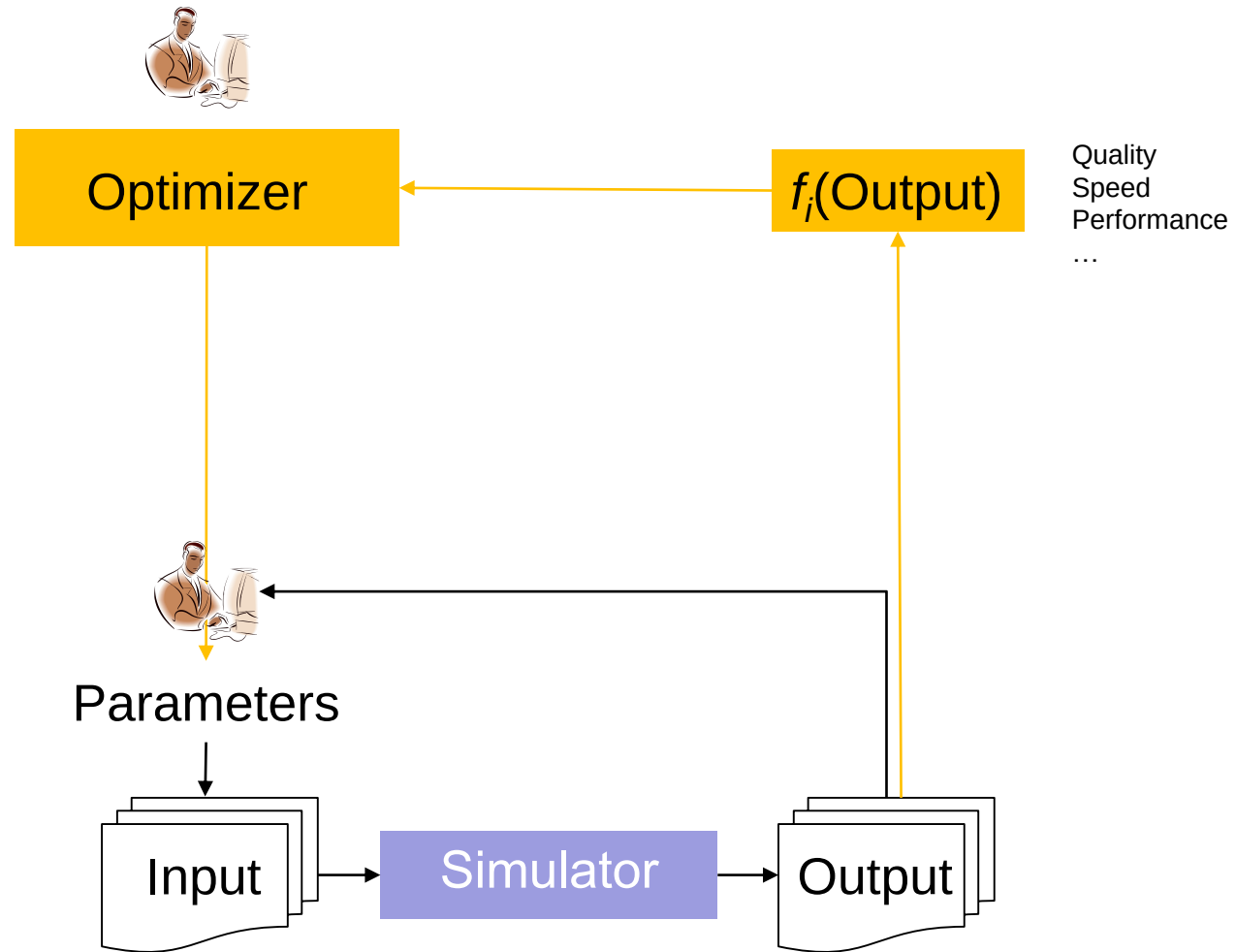
MDO Production Runs



- 13,5 kg weight reduction by advanced ES.
- Beats best so far method significantly.
- Typically faster convergence velocity of ES.
- Development time: 5 wk $\rightarrow$ 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

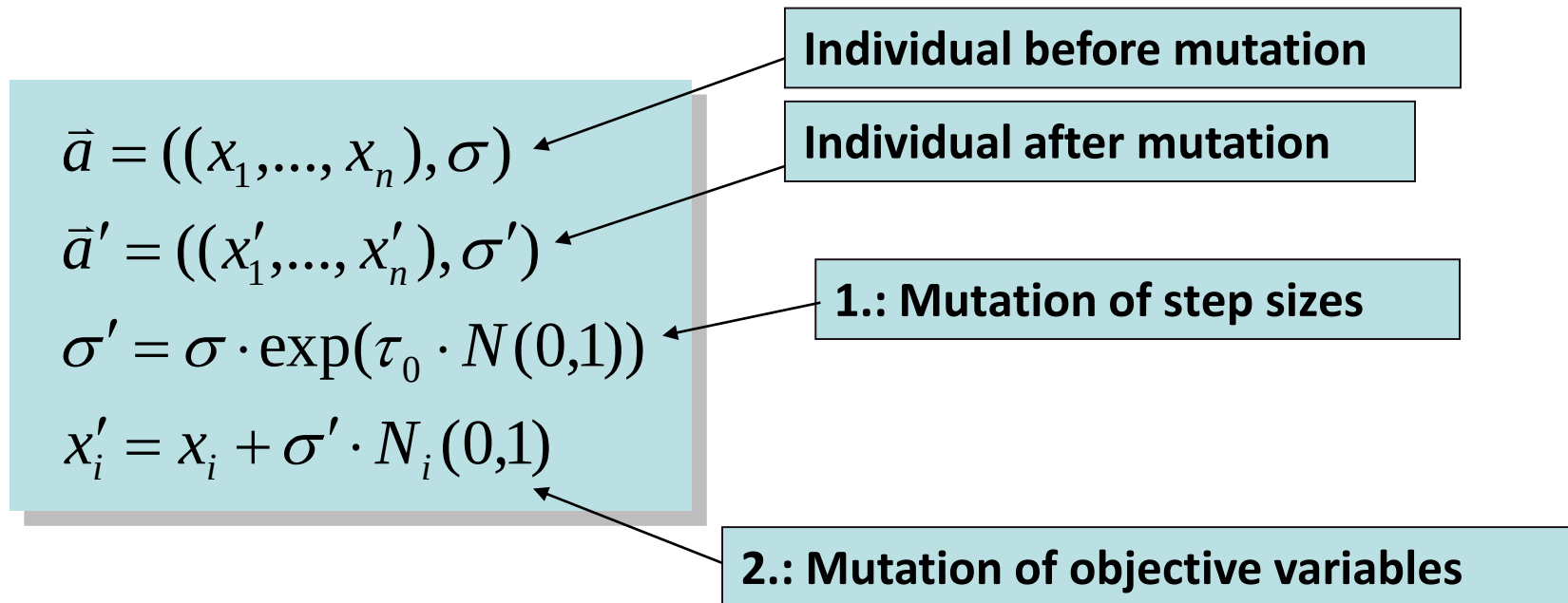
SELF-ADAPTING OPTIMIZATION ALGORITHMS

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

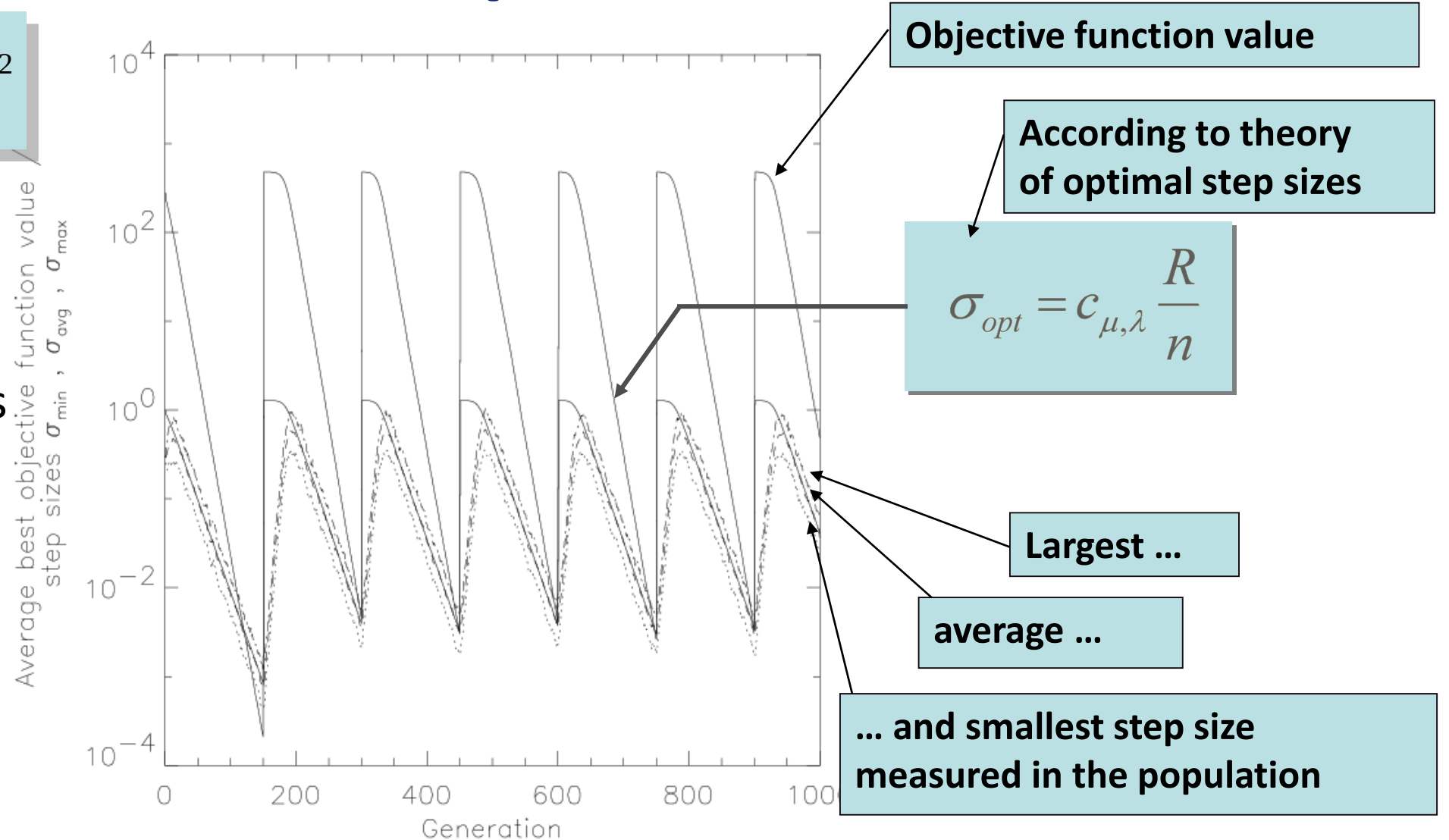
- One σ controls mutation for all x_i
- Mutation: $N(0, \sigma)$



Self-adaptation: Example

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

- Dynamic
- Changes every 150 generations



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: $C = I$, and $p_c = 0$, $p_\sigma = 0$,

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 $w_{i=1 \dots \lambda}$ such that $\mu_w = \frac{1}{\sum_{i=1}^\mu w_i^2} \approx 0.3 \lambda$

While not terminate

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

sampling

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

update mean

$$p_c \leftarrow (1 - c_c) p_c + \mathbb{1}_{\{\|p_\sigma\| < 1.5\sqrt{n}\}} \sqrt{1 - (1 - c_c)^2} \sqrt{\mu_w} y_w$$

cumulation for C

$$p_\sigma \leftarrow (1 - c_\sigma) p_\sigma + \sqrt{1 - (1 - c_\sigma)^2} \sqrt{\mu_w} C^{-\frac{1}{2}} y_w$$

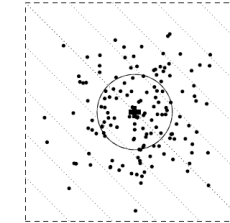
cumulation for σ

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

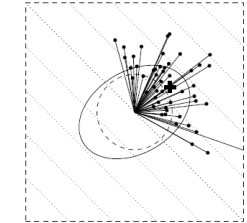
update C

$$\sigma \leftarrow \sigma \times \exp \left(\frac{c_\sigma}{d_\sigma} \left(\frac{\|p_\sigma\|}{\mathbb{E}\|\mathcal{N}(0, I)\|} - 1 \right) \right)$$

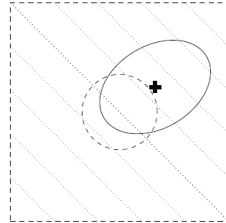
update of σ



Sampling

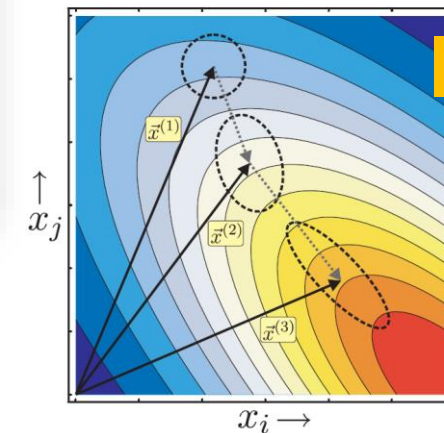


Estimation



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 ($\rightarrow$ parameters)
- Calibrating ($\rightarrow$ options)
- Programming by Optimization
 - Levels 0, 1, 2(?)

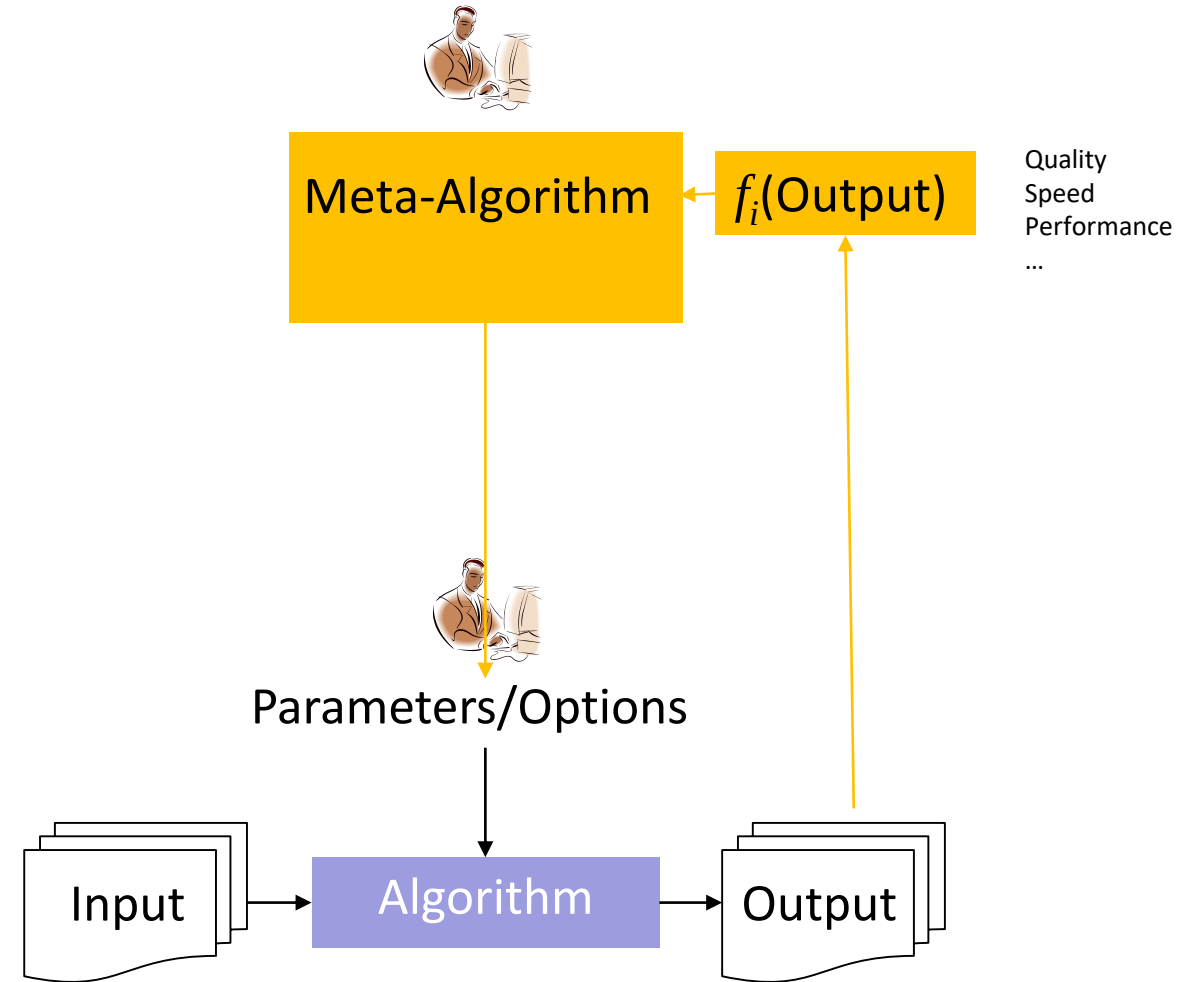
$$\theta^* = \underset{\theta \in \Theta}{\arg \max} m(\mathcal{C}(A(\theta), \Pi))$$

Statistic, e.g., mean

Set of problem instances

Algorithm

Sample of quality values over 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 ...

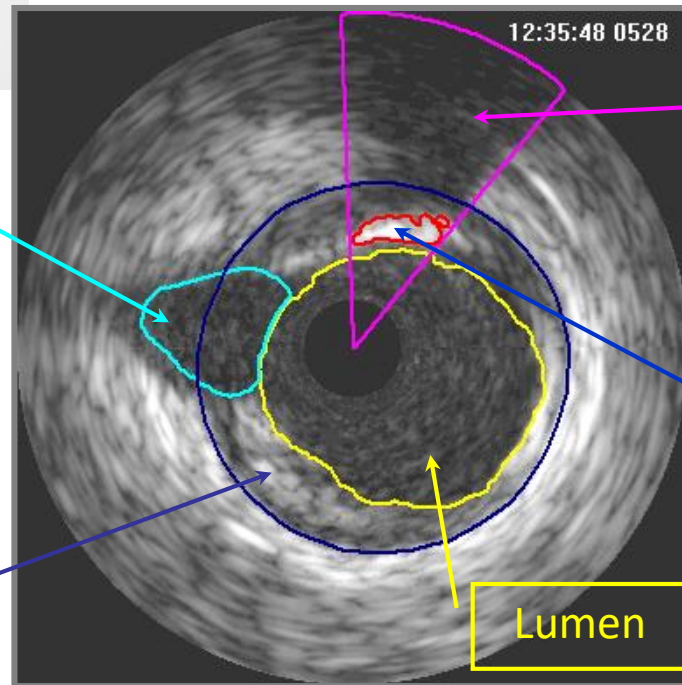
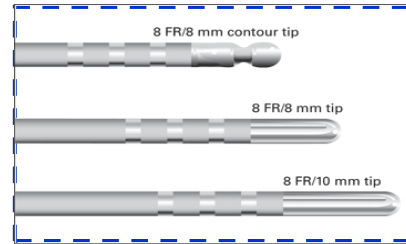
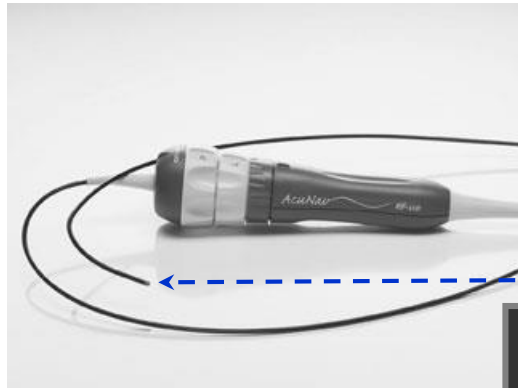
- Seven orders of magnitude improvement:
 - (7,17)-GA
 - Tournament selection, $q=20$
 - Uniform crossover, $p_c=0.421$
 - $p_m=0.00383$

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



Sidebranch

Vessel Border

Shadow

Plaque

Lumen



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
nrofoptions	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, circle}		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

dataset	default parameters		parameter solution 1		parameter solution 2		parameter solution 3		parameter solution 4		parameter solution 5	
	fitness	s.d.	fitness	s.d.	fitness	s.d.	fitness	s.d.	fitness	s.d.	fitness	s.d.
1	395.2	86.2	243.5	62%	159.8	43.5	185.4	43.0	144.8	42.0	271.0	74.8
2	400.2	109.2	239.2	54%	180.7	58.4	207.2	69.2	232.7	71.0	352.0	73.1
3	344.8	65.6	209.8	40%	203.9	70.1	164.4	49.7	183.9	80.3	327.1	55.9
4	483.1	110.6	202.7	41%	269.0	73.2	250.4	80.4	173.2	64.7	330.1	82.2
5	444.2	90.6	200.9	17%	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:

```
for  $i = 1, \dots, n_r$  do  
   $s'_i \leftarrow s_i \exp(\tau_g N_g + \tau_l N(0, 1))$   
   $r'_i \leftarrow r_i + N(0, s'_i)$ 
```

Learning rates
(local)

Discrete parameters:

Learning rates
(global)

```
end for  
for  $i = 1, \dots, n_z$  do  
   $q'_i \leftarrow q_i \exp(\tau_g N_g + \tau_l N(0, 1))$   
   $z'_i \leftarrow z_i + G(0, q'_i)$ 
```

Categorical parameters:

Mutation
probabilities

```
end for  
 $p'_i := 1 / [1 + \frac{1-p_i}{p_i} * \exp(-\tau_l * N(0, 1))]$   
for  $i \in \{1, \dots, n_d\}$  do  
  if  $U(0, 1) < p'_i$  then  
     $d'_i \leftarrow$  uniformly randomly value from  $D_i$   
  end if  
end for
```

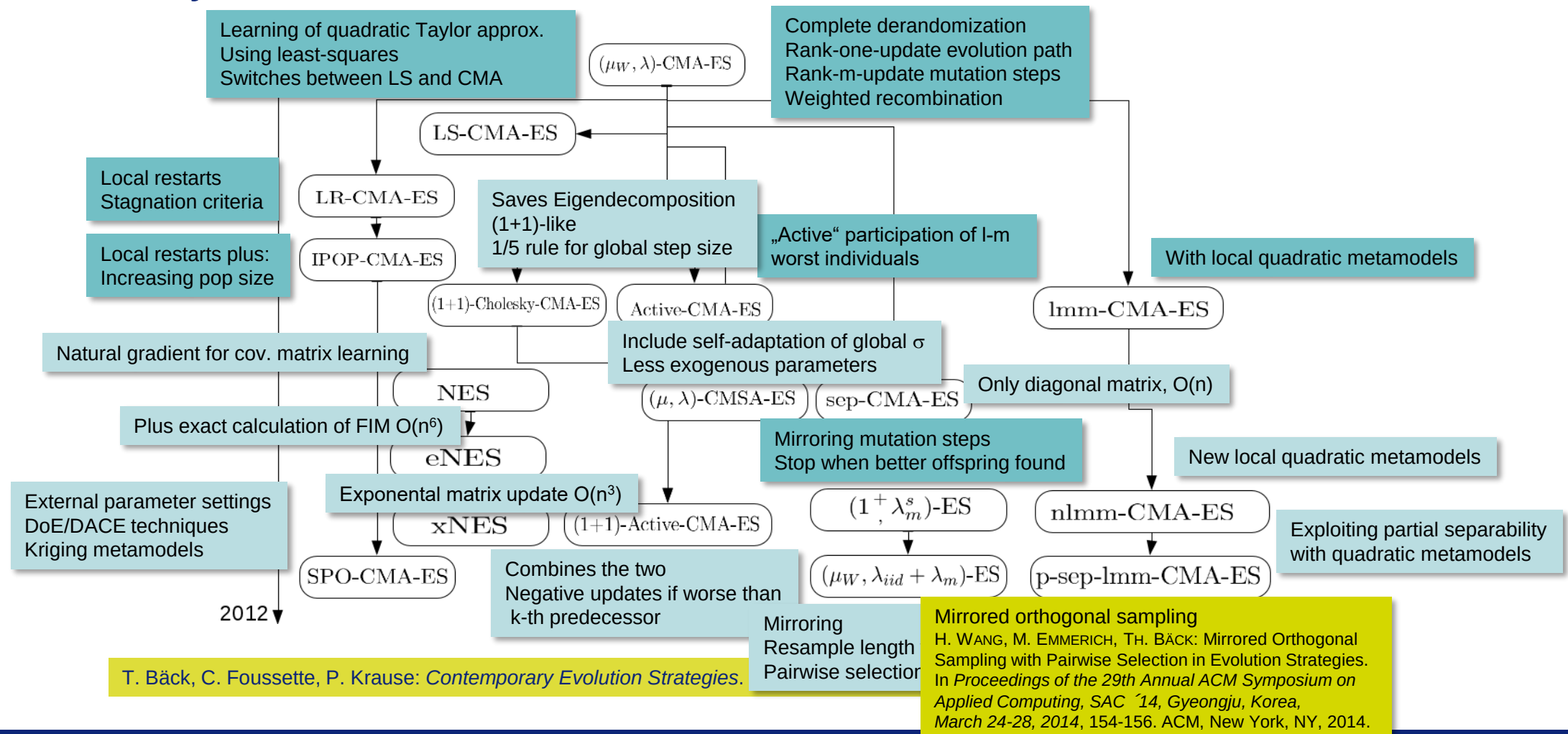
Geometrical
distribution

3.2

ALGORITHM CONFIGURATION FOR ES

Contemporary Evolution Strategies

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



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. $1/\mu$ 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

```
1: options  $\leftarrow$  which modules are active
2: // Local restart loop
3: while not terminate do
4:    $t \leftarrow 0$ 
5:    $\bar{x} \leftarrow$  randomly generated individual
6:   SetParameters(init-params)
7:   // ES execution loop
8:   while not terminate local do
9:     params  $\leftarrow$  Initialize(init-params)
10:     $\vec{x} \leftarrow$  Mutate( $\bar{x}$ , options)           // Sampler, Threshold
11:     $\vec{f} \leftarrow$  Evaluate( $\vec{x}$ , options)       // Sequential
12:     $P^{(t+1)} \leftarrow$  Select( $\vec{x}$ ,  $\vec{f}$ , options)  // Elitism, Pairwise
13:     $\bar{x} \leftarrow$  Recombine( $P^{(t+1)}$ , options) // Weights
14:    UpdateParams(params, options)           // Active, TPA
15:     $t \leftarrow t + 1$ 
16:  end while
17:  AdaptParams(init-params)                 // (B)IPOP
18: 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 Selection
6. 1/5 rule

7.
8.

11.

4,608 possible

Meta-algorithm:

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

Algorithm 1 Modular CMA-ES Framework

- 1: options $\leftarrow$ which modules are active
- 2: *// Local restart loop*
- 3: **while** not terminate **do**

4.608 Algorithms?

- 17: AdaptParams(init-params) *// (B)IPOP*
- 18: **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 Selection
6. 1/5 rule

- 7.
- 8.

11.

4,608 possible

Meta-algorithm:

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

Algorithm 1 Modular CMA-ES Framework

- 1: options $\leftarrow$ which modules are active
- 2: *// Local restart loop*
- 3: **while** not terminate **do**

Combinatorial Algorithmics

- 17: AdaptParams(init-params) *// (B)IPOP*
- 18: **end while**

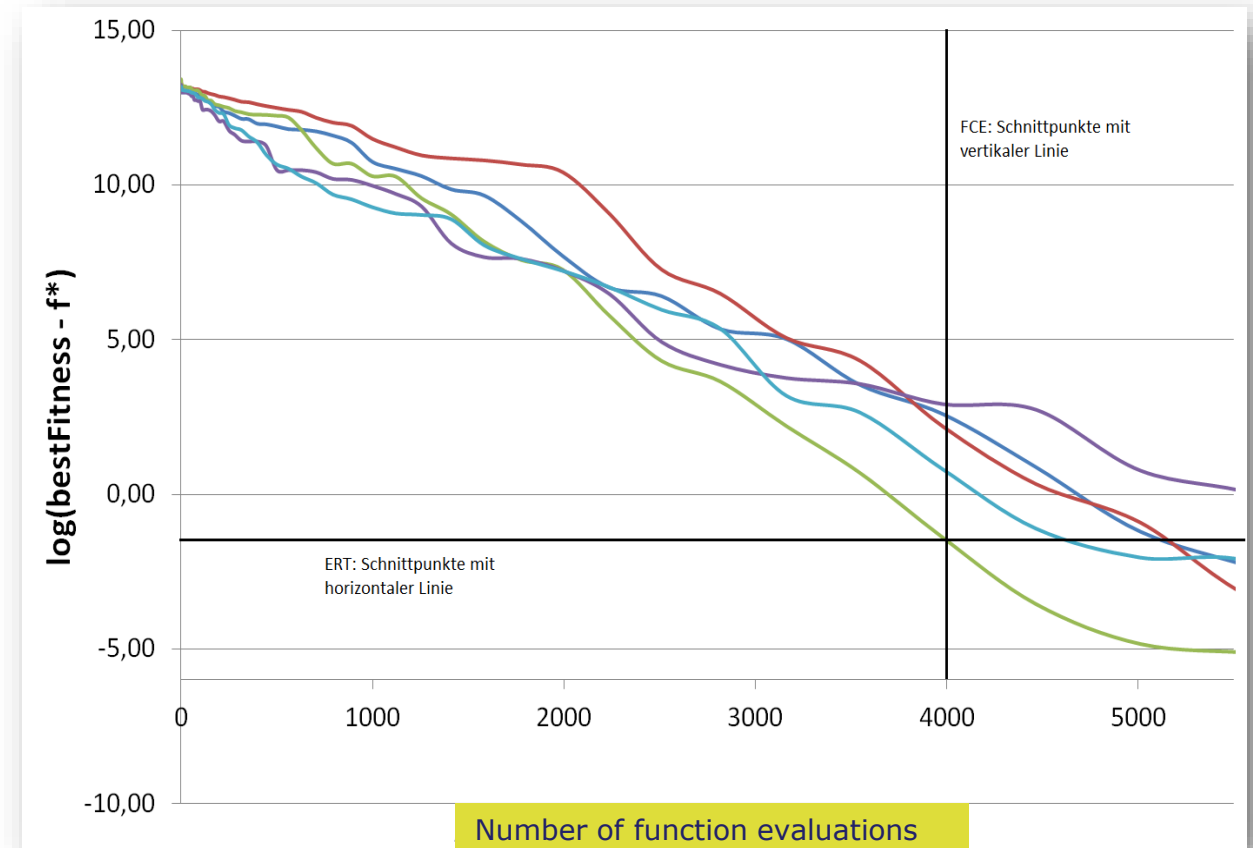
S. VAN RIJN, H. WANG, M. VAN LEEUWEN, TH. BÄCK: Evolving the Structure of Evolution Strategies. In: *IEEE Symposium Series on Computational Intelligence, Athens, Greece, Dec. 6-9, 2016*. IEEE Press, Piscataway, NJ, 2016.

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]

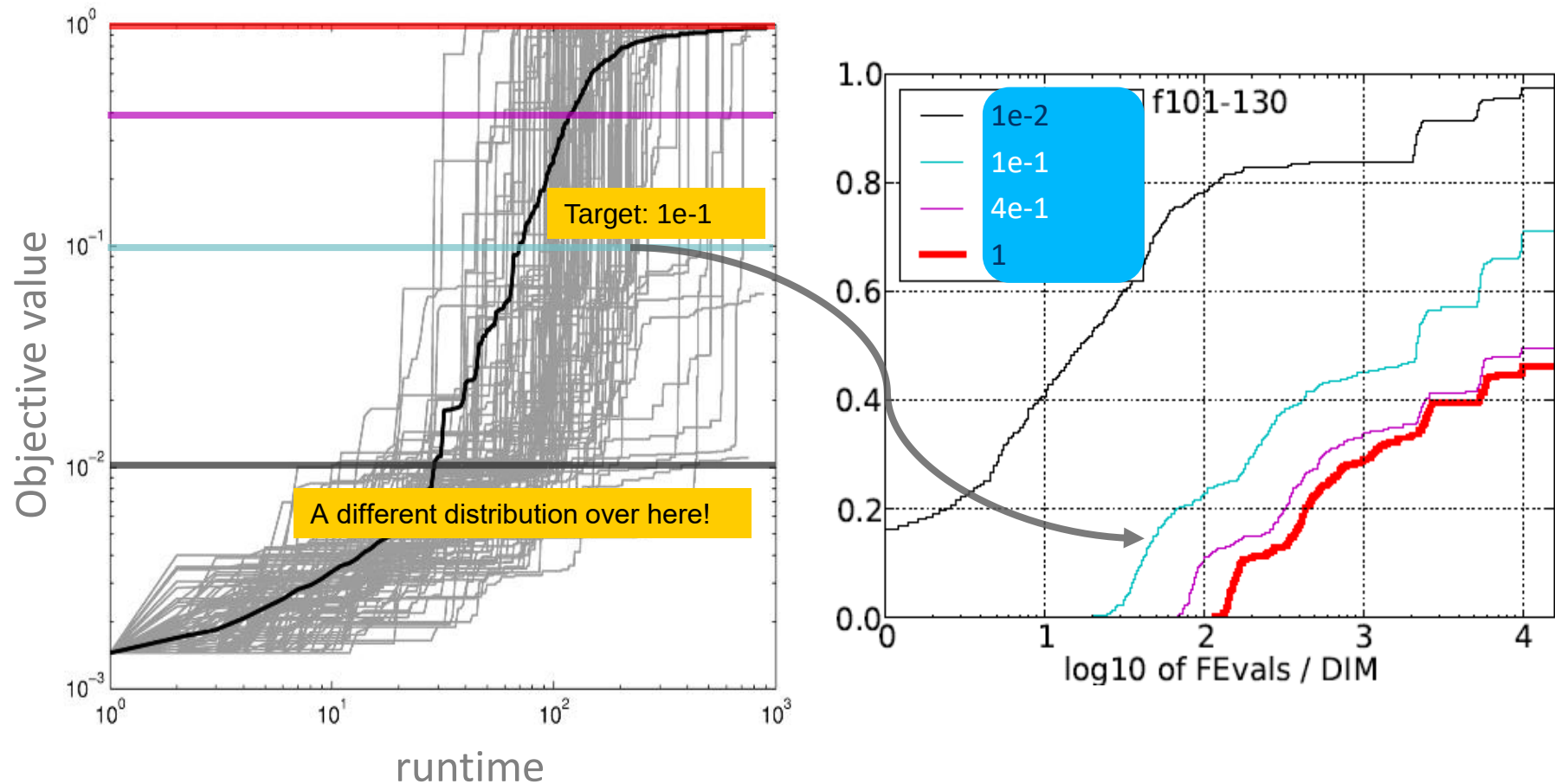
Efficiency Measures

- 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

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

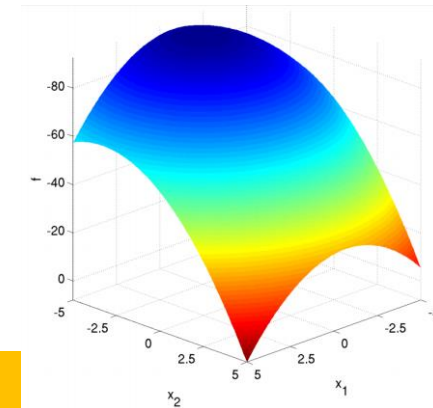
Top-10 from Brute Force enumeration

[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)

Mirrored + orthogonal sampling + $1/\mu$ 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

[10000000002]

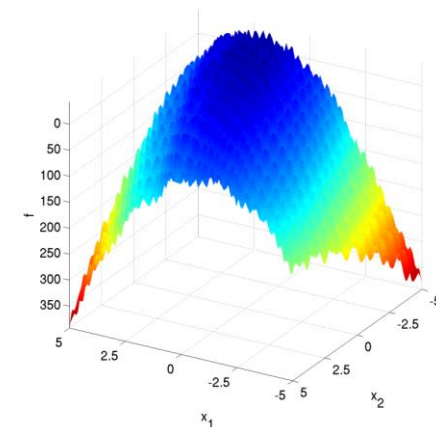
Top-10 from Brute Force enumeration

[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
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

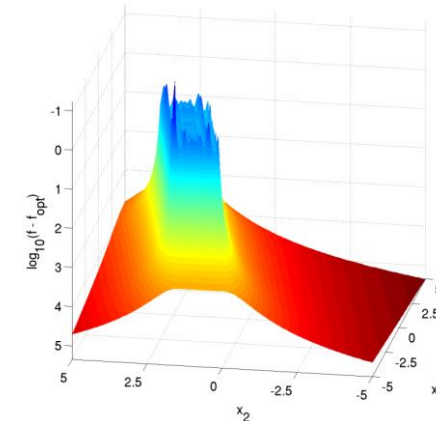
[00100001000]

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

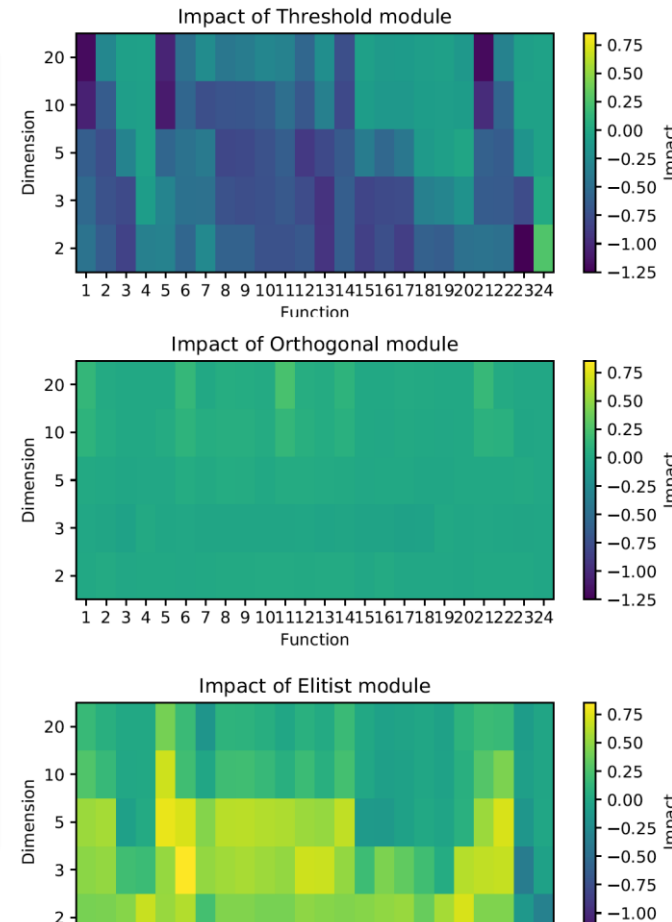
Mining the Results

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

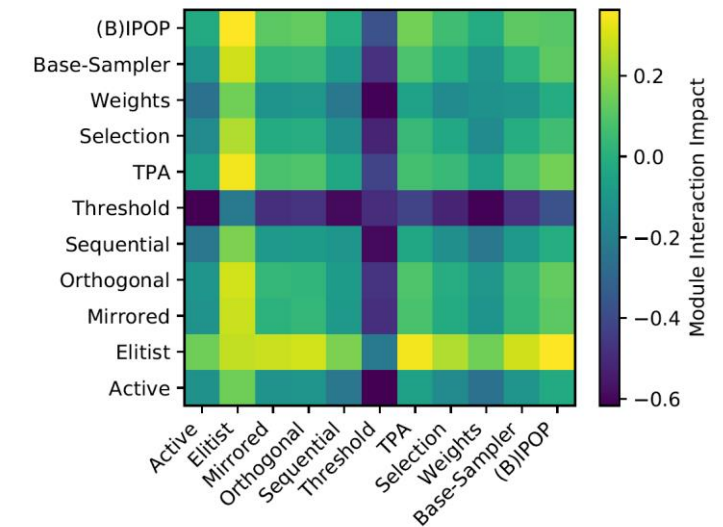


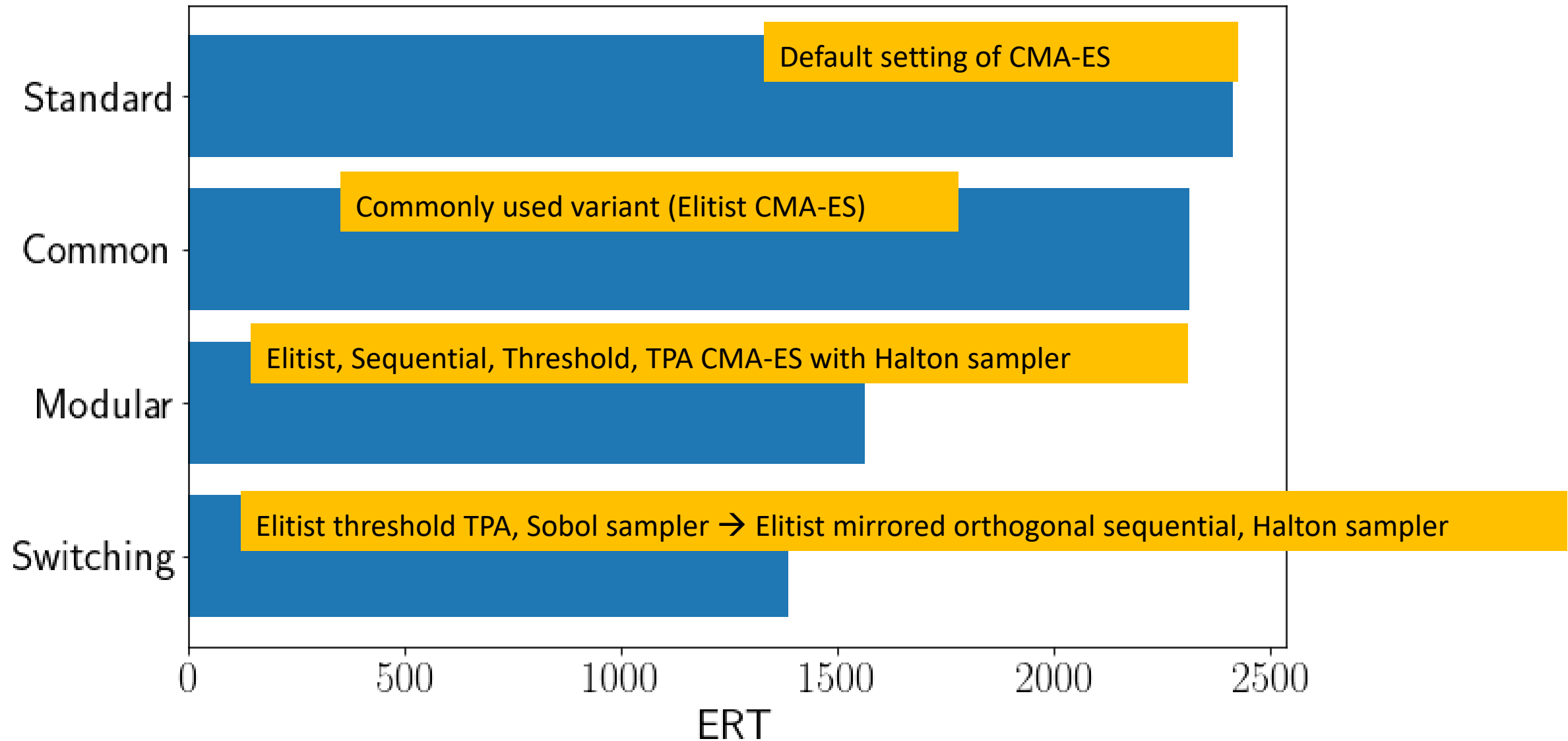
Figure 3: Aggregate Module Interaction Impact. Displayed impact scores are calculated as a mean over all 120 experiments. Each score indicates the difference in q -score between the set of configurations with two modules x and y active, versus the remaining configurations. The heatmap is symmetrical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

3.3

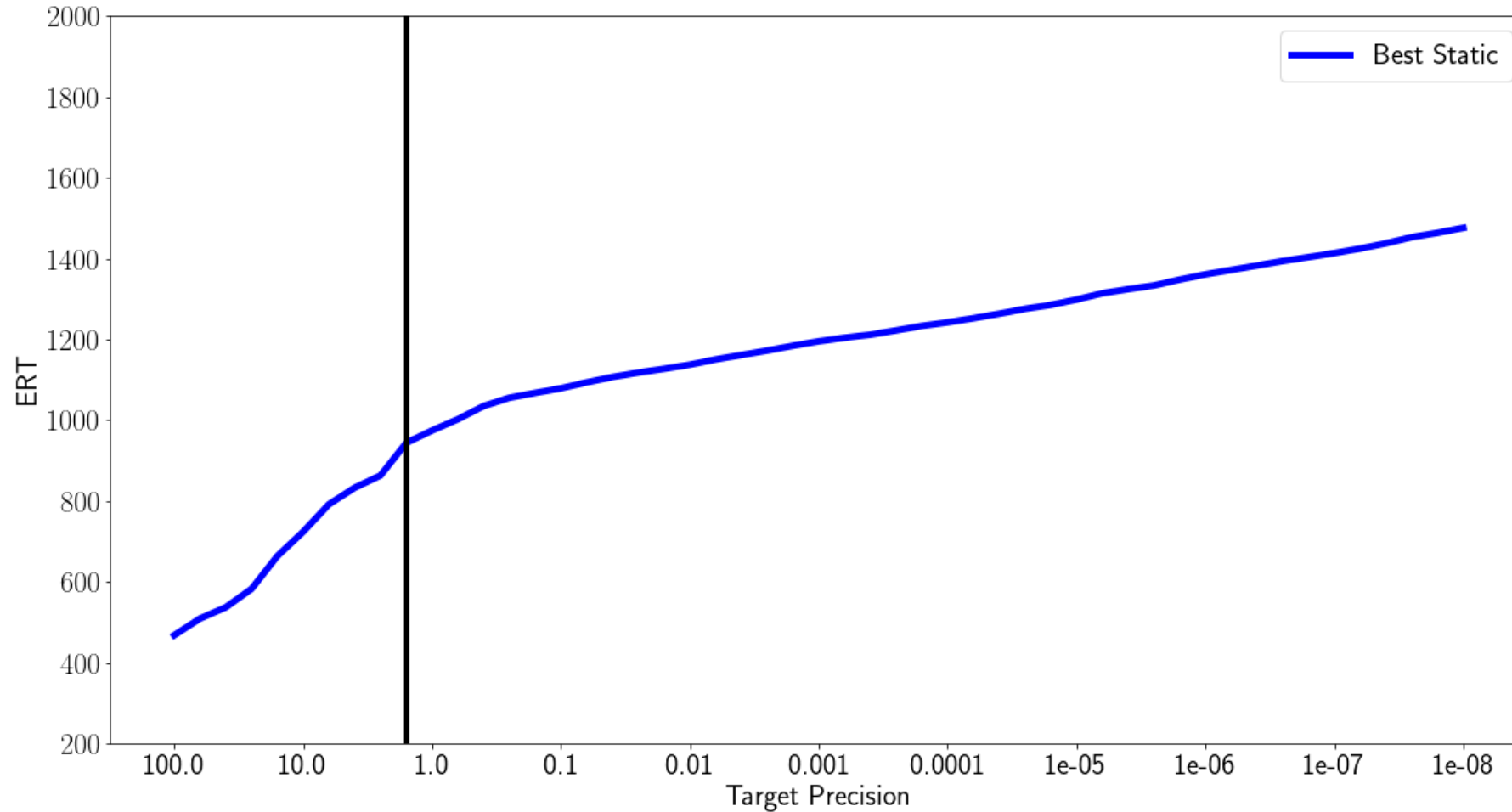
ONLINE SELECTION OF VARIANTS

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).

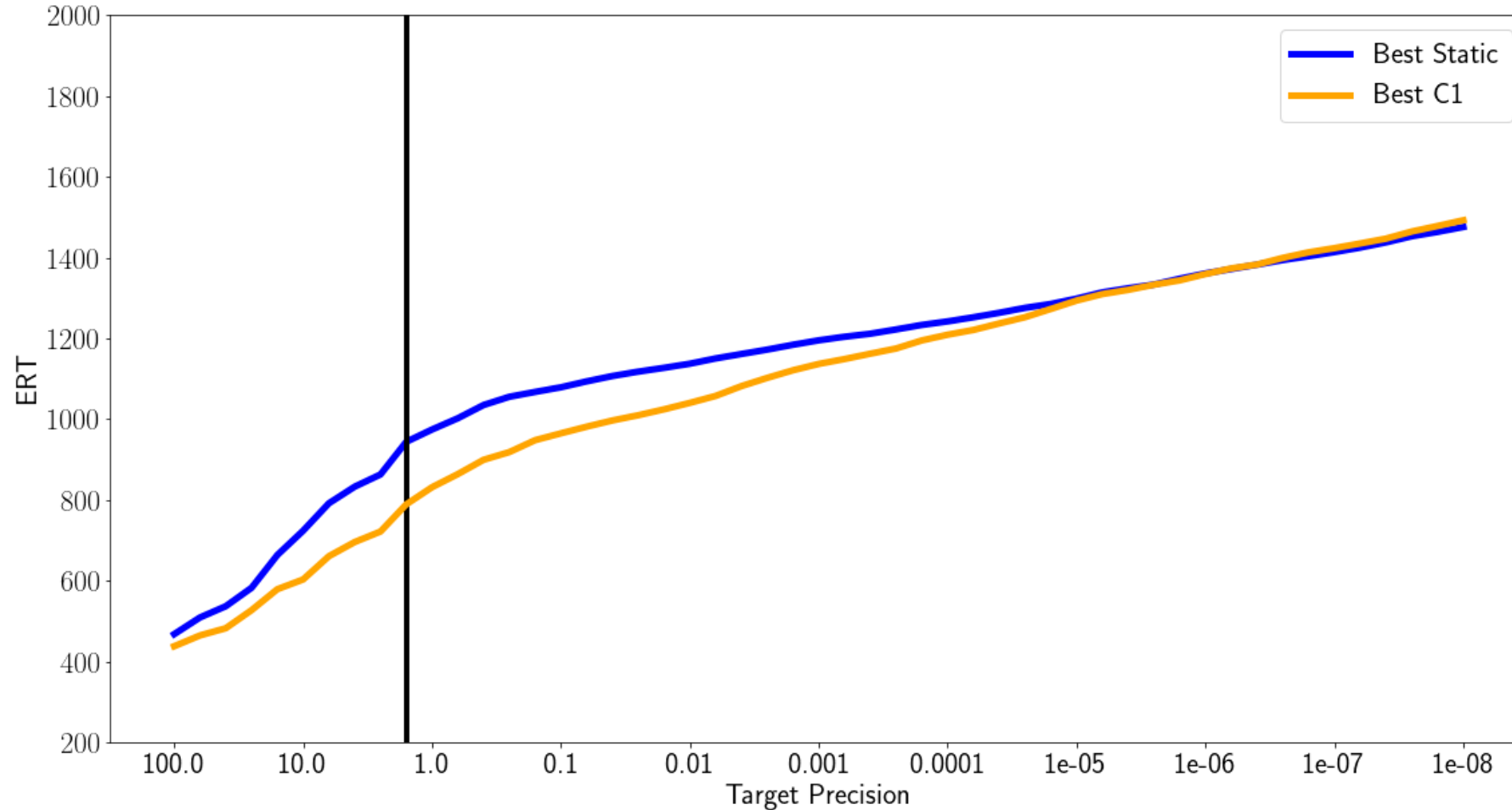
Results (F10)



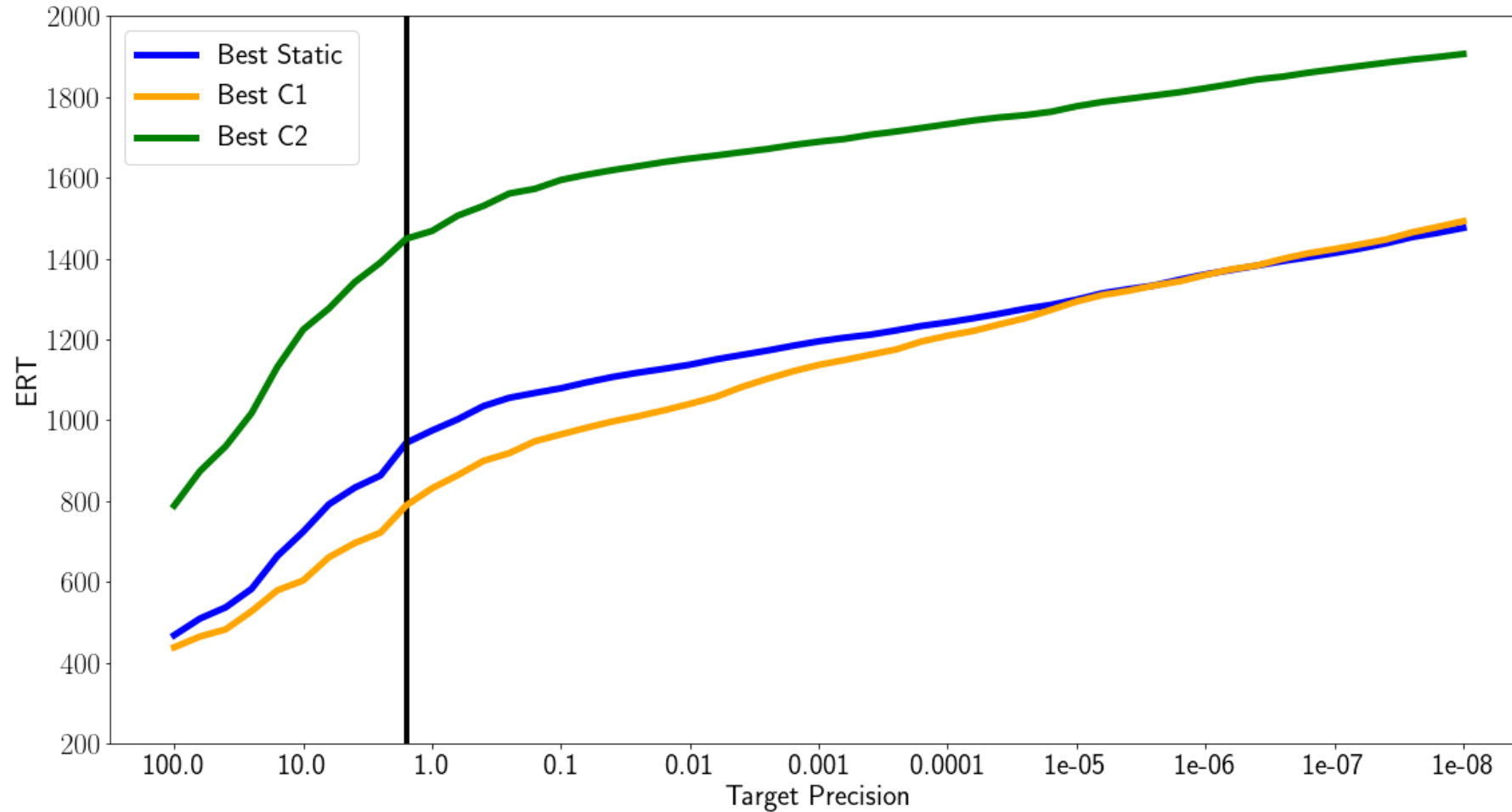
Idea Behind Configuration Switching



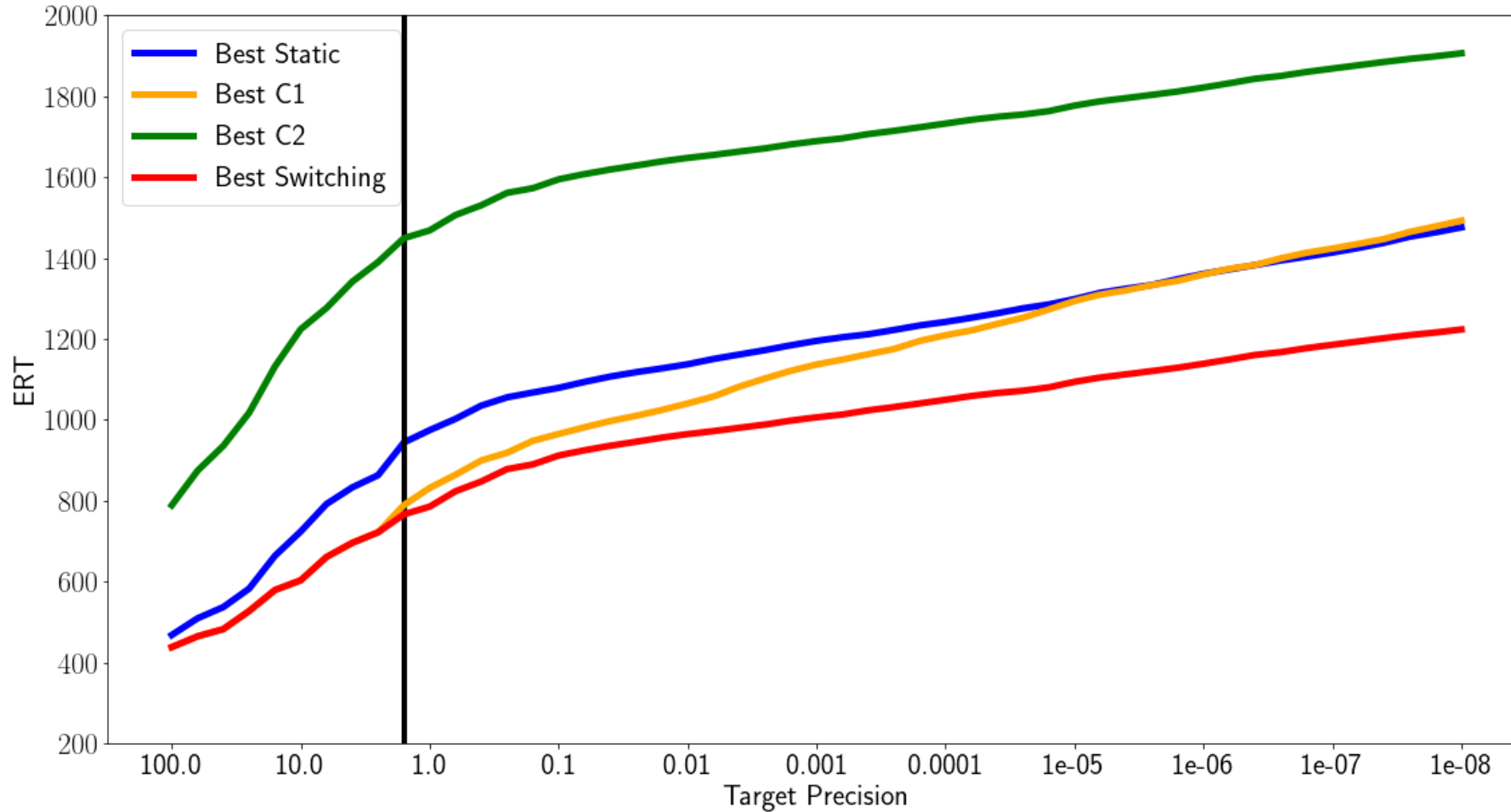
Idea Behind Configuration Switching



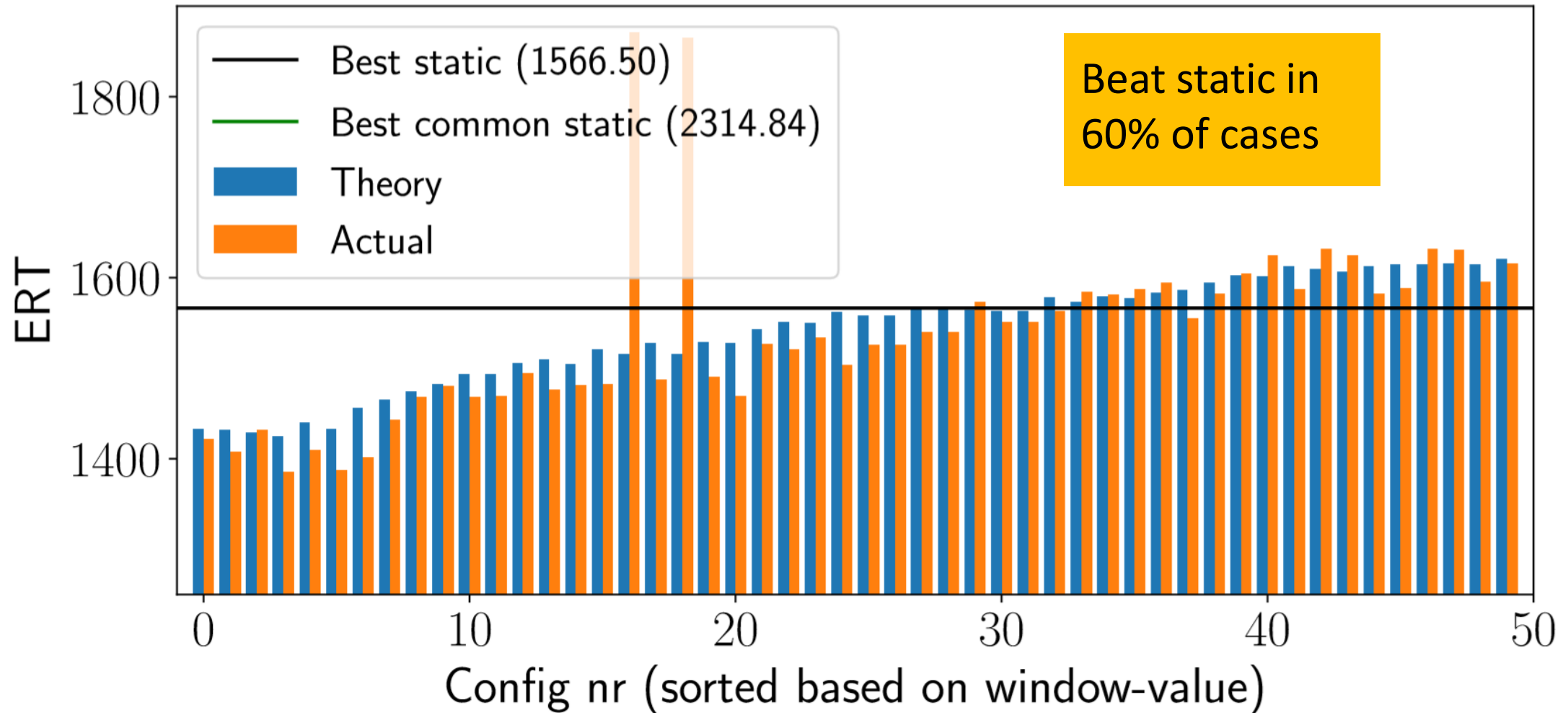
Idea Behind Configuration Switching



Idea Behind Configuration Switching



Zooming in on F10



Zooming in on F10

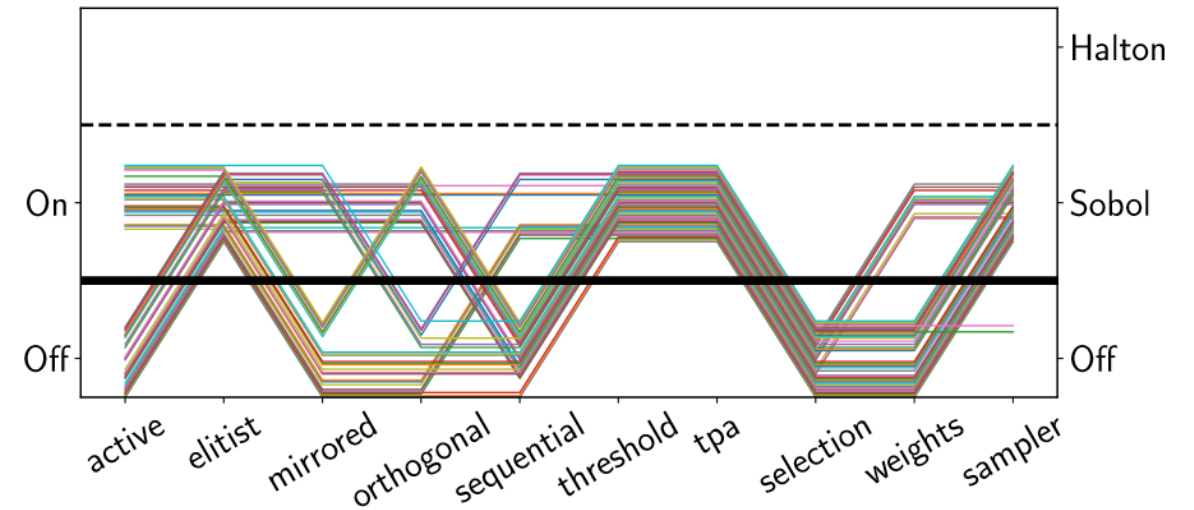
- Visually obvious difference between C_1 and C_2

- Define Maximum module difference:

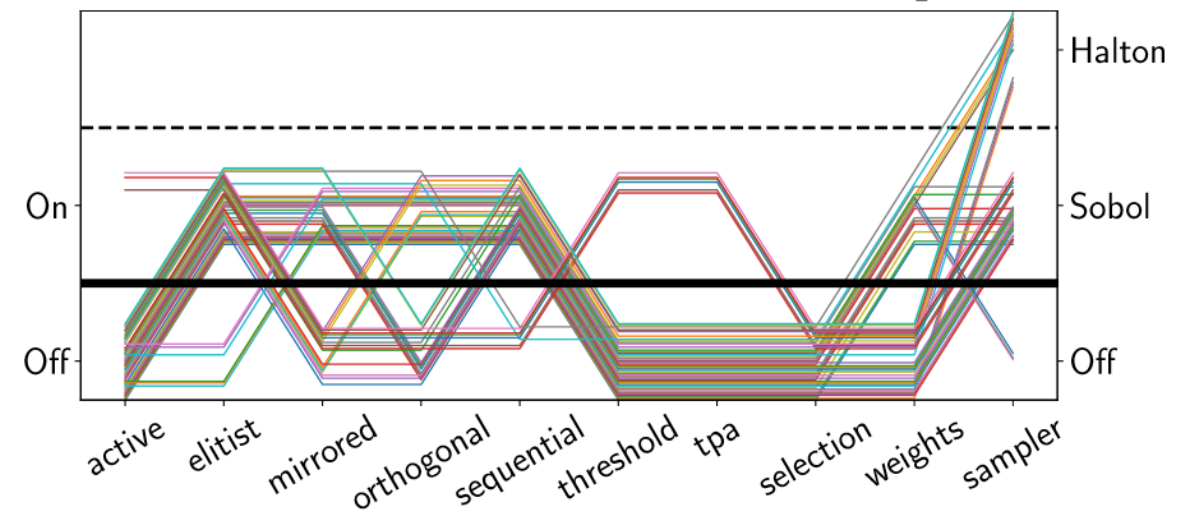
$$\frac{100}{n} \max_{Module\ m} \left| \sum_1^n m(C_1(i)) - \sum_1^n m(C_2(i)) \right|$$

- Here: 88%
- Selection based on behavior of configuration

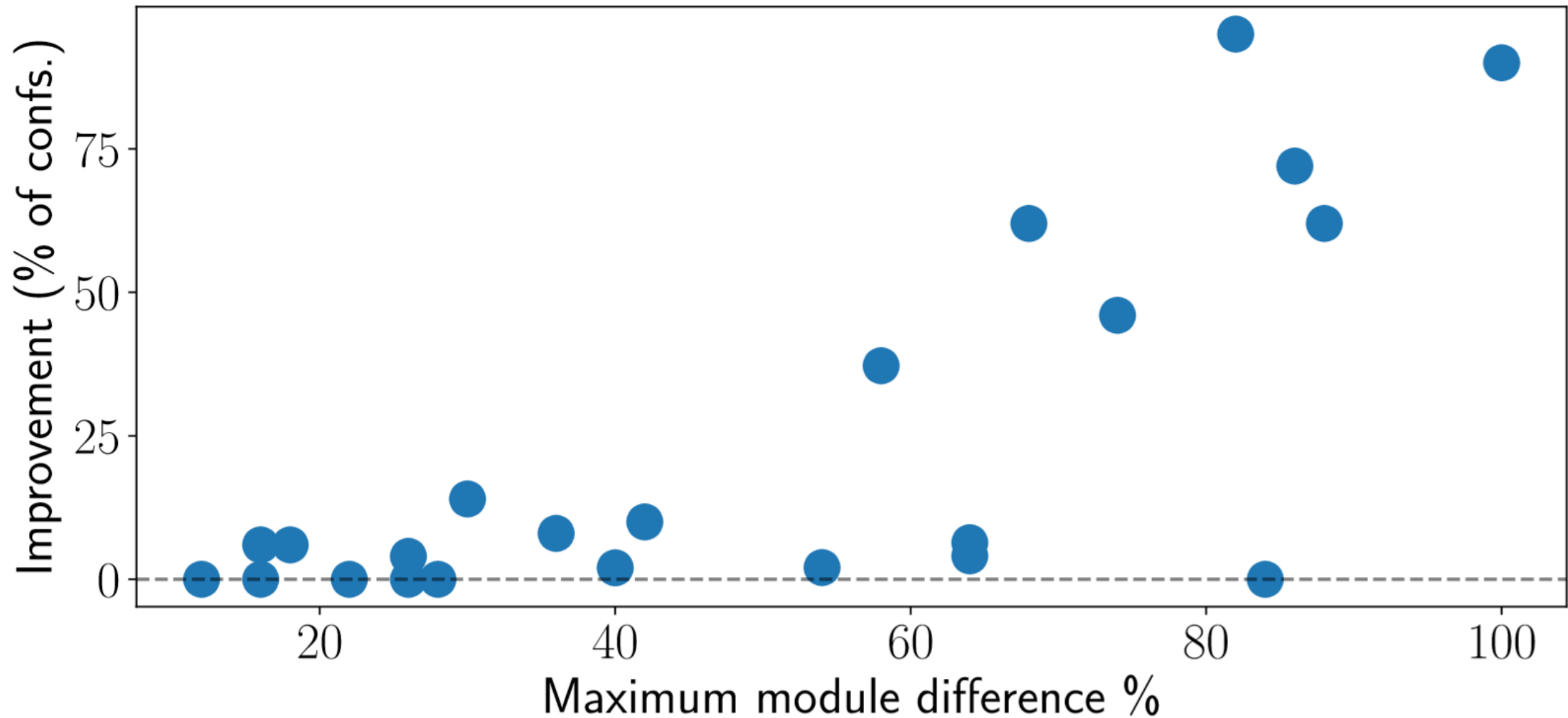
F10: Combined module activation for C_1



F10: Combined module activation for C_2



General Trend



4

CONCLUDING

Concluding Remarks

- Evolutionary Algorithms are powerful general-purpose optimizers
- Can be used for algorithm configuration as well
- Combinatorial algorithms 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)



Volkswagen



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

Thomas Bäck (Natural Computing Group)



Volkswagen



Optimization:

- Evolutionary Computation
- Optimization of Optimizers
- Optimizer Configuration
- Efficient Global Optimization
- Bayesian Optimization
- Genetic Programming

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



SORBONNE
UNIVERSITÉ



Tel-Hai
College

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. Laat
 - Honorary Chairs:
 - H.-P. Schwefel
 - G. Rozenberg



PPSN 2020

Leiden, The Netherlands
September 5 - 9, 2020

PPSN 2020 - FIRST CALL FOR PAPERS

16TH INTERNATIONAL CONFERENCE ON PARALLEL PROBLEM SOLVING FROM NATURE

The Sixteenth International Conference on Parallel Problem Solving from Nature (PPSN XVI) will be held in Leiden, Netherlands, September 5-9, 2020. Leiden University and the Leiden Institute of Advanced Computer Science (LIACS) are proud to host the 30th anniversary of PPSN.

PPSN brings together researchers and practitioners in the field of Natural Computing: The study of computing approaches which are gleaned from natural models. The field includes, but is not limited to, areas such as Amorphous Computing, Artificial Life, Artificial Ant Systems, Artificial Immune Systems, Artificial Neural Networks, Cellular Automata, Evolutionary Computation, Swarm Computing, Self-Organizing Systems, Chemical Computation, Molecular Computation, Quantum Computation, and Machine Learning and Artificial Intelligence approaches using Natural Computing methods.

PPSN XVI will feature workshops and tutorials covering advanced and fundamental topics in the field of Natural Computing, as well as algorithm competitions. The keynote talks will be given by world-renowned researchers in their fields.

Following PPSN's unique tradition, all accepted papers will be presented during poster sessions and will be included in the proceedings. The proceedings will be published in the Lecture Notes in Computer Science (LNCS) series by Springer. Prospective authors are invited to contribute their high-quality original results in the field of natural computation as papers of no more than 12 pages plus references. The format follows Springer Verlag LNCS style.

Leiden is just 15 minutes by train from the 3rd largest airport in Europe, Schiphol Amsterdam (AMS) – which can be reached by direct flights from more than 300 destinations world-wide.

We are very much looking forward to seeing you at PPSN XVI in Leiden!

Workshop/Tutorial/Competition proposals
November 13, 2019

Workshop/Tutorial/Competition notification
November 27, 2019

Abstract submission
March 25, 2020

Paper submission
April 1, 2020

Author notification
May 11, 2020

Camera ready
June 1, 2020

Conference
September 5 - 9, 2020

IMPORTANT DATES

GENERAL CHAIRS Mike Preuss Thomas Bäck	PROCEEDINGS CHAIRS Hao Wang André Deutz	TUTORIALS CHAIR Ofer Shir	WORKSHOP CHAIR Anna Esparcia-Alcazar	FINANCIAL CHAIR Felix Wittleben
PROGRAM COMMITTEE CHAIRS Michael Emmerich Heike Trautmann Carola Doerr	COMPETITIONS CHAIR Vanessa Volz HONORARY CHAIRS Hans-Paul Schwefel Grzegorz Rozenberg	KEYNOTE CHAIR Aske Plaatt	INDUSTRIAL LIAISON CHAIR Bernhard Sendhoff	LOCAL TEAM Roshny Kohabir Jayshri Murli

ppsn2020.liacs.leidenuniv.nl



The Natural Computing Group

