Theory of Iterative Optimization Heuristics: From Back-Box Complexity over Algorithm Design to Parameter Control

Carola Doerr
CNRS researcher at LIP6, Sorbonne University, Paris, France

Soutenance d'Habilitation à Diriger des Recherches
December 18, 2020

Soutenance d'Habilitation à Diriger des Recherches

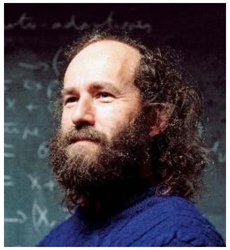
Jury members:

- Laetitia Jourdan, Université de Lille, France (reviewer)
- Jonathan E. Rowe, The University of Birmingham, UK (reviewer)
- Carsten Witt, Technical University of Denmark, Denmark (reviewer)
- Carlos Artemio Coello Coello, CINVESTAV-IPN, Mexico (examiner)
- Christoph Dürr, Sorbonne Université, France (examiner)
- Günter Rudolph, TU Dortmund University, Germany (examiner)
- Marc Schoenauer, INRIA, France (examiner)
- Lothar Thiele, ETH Zürich, Switzerland (examiner)










- Time line:
  - ~45 minutes: scientific résumé
  - questions by the jury members
    happy to take questions from the audience during the coffee break
  - jury-internal discussion // virtual coffee break
    different zoom location // here

Personal Background

- 2007: Diplom (≅ Master's degree) in Mathematics
  University of Kiel, Germany
- 2007-2009 (on leave until 2012):
  Business Consultant with McKinsey & Company
  Munich, Germany
- 2010-2011: PhD in Computer Science
  Max Planck Institute for Informatics
  and Saarland University, Germany
  Algorithms and Complexity Group of Kurt Mehlhorn
- 2012-2013: PostDoc
  Max Planck Institute
  LIAFA (now IRIF) at Paris Diderot (now Université de Paris)
- Since 10/2013: CNRS researcher
  LIP6, Sorbonne Université
  Operations Research team (RO)

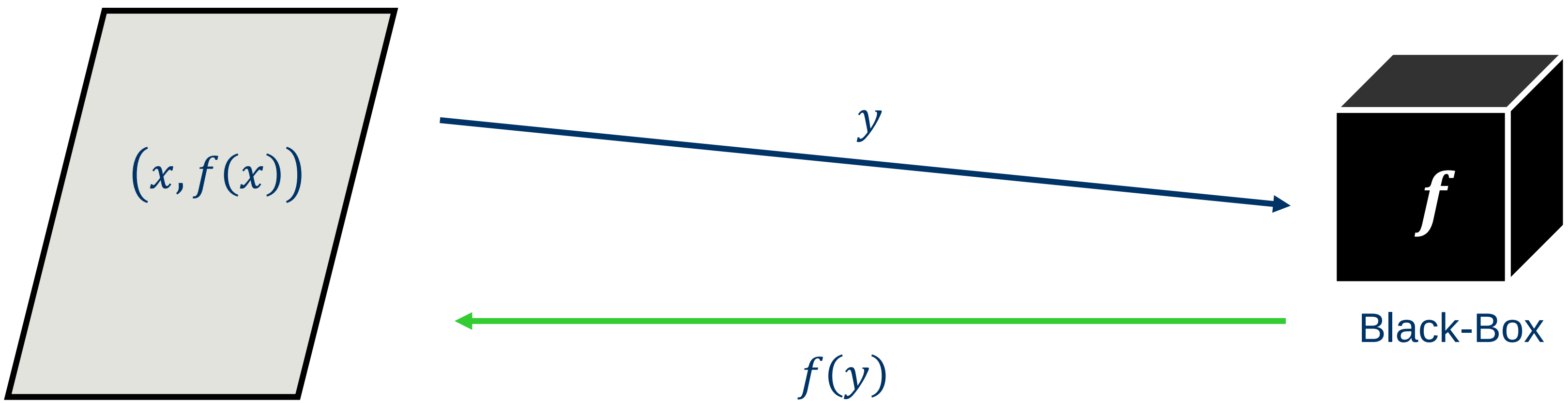
Student Supervision and Teaching Activities

- **PostDocs**
  - Hao Wang 01/2020-08/2020
    now Assistant Professor at LIACS, Leiden University
  - Martin Krejca 01/2021-10/2022
- **PhD students**
  - Diederick Vermetten (Leiden University, 01/2020-)
    co-supervising with Thomas Bäck
  - Quentin Renau (CIFRE Thales, 02/2019-)
    co-supervising with Benjamin Doerr and Johann Dreo
  - Anja Jankovic (Sorbonne Université, 10/2018-09/2021)
    main supervisor
  - Furong Ye (Leiden University, 10/2017-09/2021)
    co-supervising with Thomas Bäck
  - Jing Yang (École Polytechnique, 10/2015-09/2018)
    co-supervised with Benjamin Doerr
- 14 Master students, 1 Bachelor student, 1 PhD interns
- Responsible for the course MPRI 2-24-2 on *Solving Optimization Problems with Search Heuristics* (with Christoph Dürr, since 2015)

Black-Box Optimization

Goal: maximize f

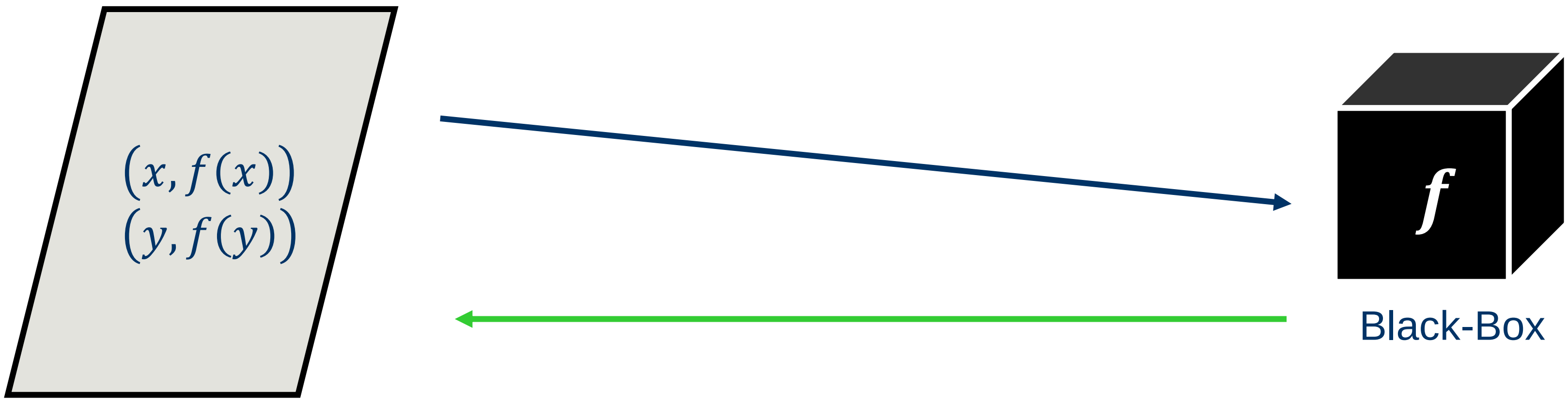
find x^* with $f(x^*)$ as large as possible



Only need information on the decision space (i.e., the domain of f)

- number of decision variables
- their type or range
- (constraints)

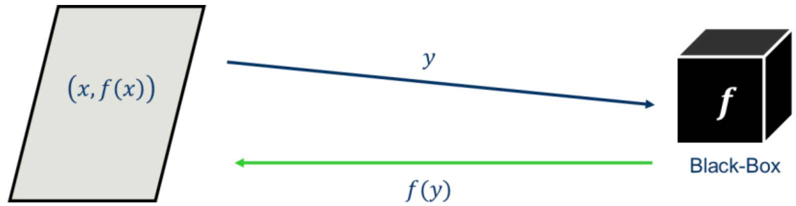
Black-Box Optimization



Sampling-Based Optimization Heuristics

Iterative Optimization Heuristics (IOHs)	One-Shot Optimization Heuristics (non-adaptive sampling)
▪ sample solution candidates ▪ evaluate them ▪ adjust your sampling strategy	▪ sample solution candidates at once ▪ evaluate all of them ▪ recommend a final solution

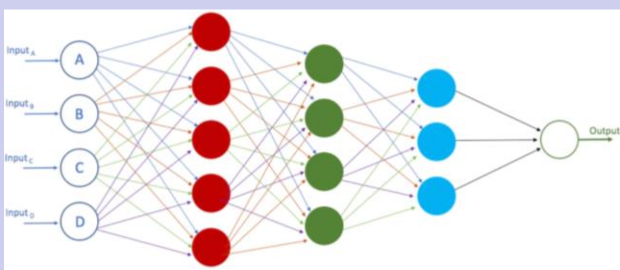
Sampling-Based Optimization Heuristics

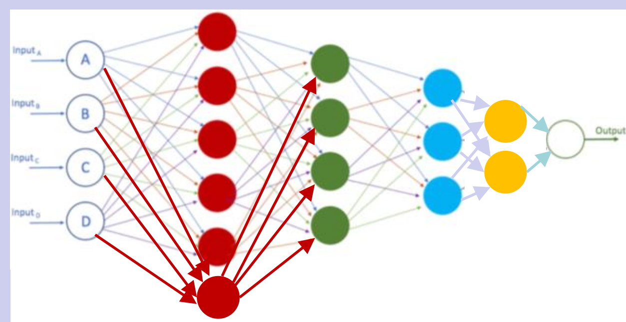


- ✓ Broadly applicable and easy to re-use:
  sampling of solution candidates
  vs. "classic" optimization: *construct* solutions

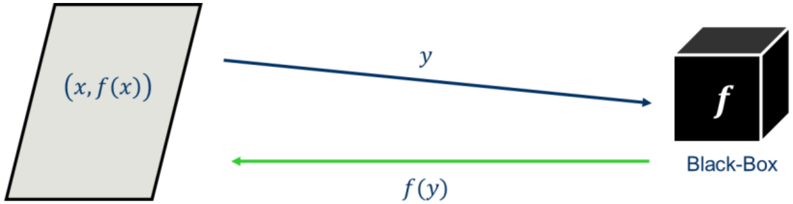
- ✓ Avoids explicit problem formulation $f: S \rightarrow \mathbb{R}$

very important feature, since we often do not have such an explicit description! (*black-box problem*)



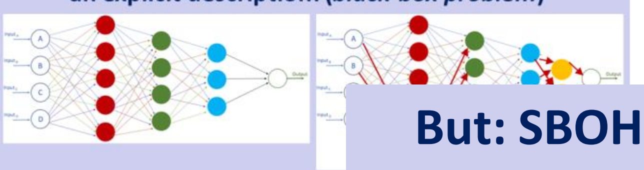


Sampling-Based Optimization Heuristics



- ✓ Broadly applicable and easy to re-use:
  sampling of solution candidates
  vs. "classic" optimization: *construct* solutions
- ✓ Avoids explicit problem formulation $f: S \rightarrow \mathbb{R}$

very important feature, since we often do not have such an explicit description! (*black-box problem*)

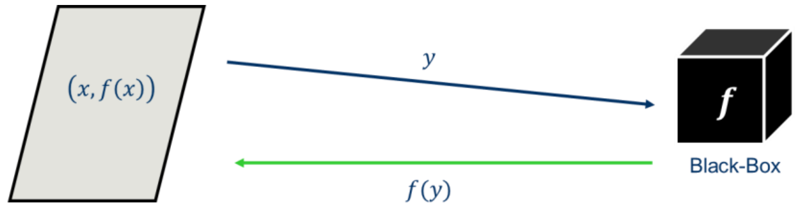


But: SBOHs can be algorithms of choice even when problem is "*grey-*" or even "*white-box*"

Example: Low Autocorrelation Binary Sequence

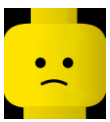
$$E(S) = \sum_{k=1}^{N-1} C_k^2(S) \qquad C_k(S) = \sum_{i=1}^{N-k} s_i s_{i+k}$$

Sampling-Based Optimization Heuristics



✓ Broadly applicable and easy to re-use:
sampling of solution candidates
vs. "classic" optimization: *construct* solutions

✓ Avoids explicit problem formulation $f: S \rightarrow \mathbb{R}$


Not so well understood. The sheer amount of design choices puts a high burden on the users of SBOHs (``Achille's heel of Evolutionary Computation" [1])

Key question: Given a problem (instance) P, which algorithm should we use?

[1] Álvaro Fialho, Luis Da Costa, Marc Schoenauer, Michèle Sebag. *Analyzing bandit-based adaptive operator selection mechanisms*. Annals of Mathematics and Artificial Intelligence, 2010

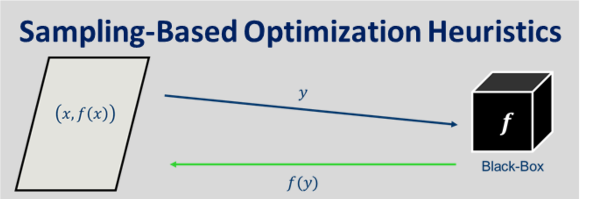
Empirical Approach

- real-world instances
- everything you can implement
- exact numbers
- typically easy to set up
- only a finite number of instances of bounded size
  → representative?
- only tells you numbers
- depends on implementation
- only single algorithms

Mathematical Approach

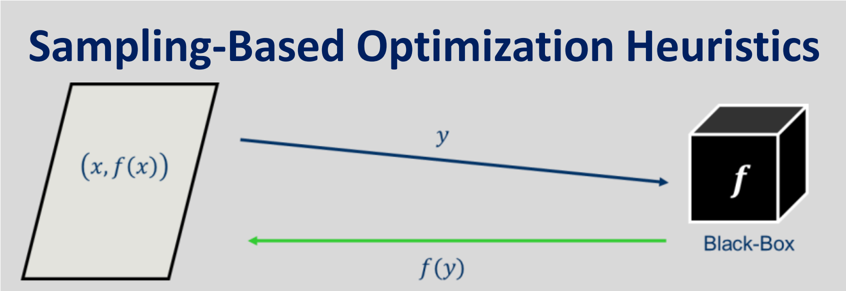
- only models for real-world instances
- limited scope, e.g., (1+1) EA
- limited precision, e.g., $O(n^2)$
- finding proofs can be difficult
- results hold for whole classes of algorithms
  → guarantee!
- proof tells you the reason
- implementation independent
- lower bounds (= performance limits)

Complementary Results



- What is the best possible performance that a SBOH can achieve for a given problem f?
- Performance measure: optimization time
  $T(A, f)$: number of evaluations needed to find an optimal solution
- Objective: $\inf_A \mathbb{E}[T(A, f)]$
- Black-box complexity of $\mathcal{F}$: $\boxed{\mathrm{BBC}(\mathcal{F}) = \inf_A \sup_{f \in \mathcal{F}} \mathbb{E}[T(A, f)]}$
- 2 Approaches to determine $\mathrm{BBC}(\mathcal{F})$:
  - Algorithm Design and Analysis: upper bound for $\mathrm{BBC}(\mathcal{F})$
  - Complexity Theory: lower bound for $\mathrm{BBC}(\mathcal{F})$

When upper and lower bound match,
we know for sure that we can stop searching for better algorithms



- $\mathcal{A}$-black-box complexity of $\mathcal{F}$: $\inf_{A \in \mathcal{A}} \sup_{f \in \mathcal{F}} \mathbb{E}[T(A,f)]$
- Examples for $\mathcal{A}$:
  - Poly-time algorithms
  - non-adaptive algorithms (*one-shot optimizers*)
  - deterministic algorithms
  - restricted memory
  - type of distributions from which we sample solution candidates
  - …

$\mathrm{BBC}(\mathcal{F},\mathcal{A}) \leq \mathrm{BBC}(\mathcal{F},\mathcal{B})$ for $\mathcal{B} \subseteq \mathcal{A}$

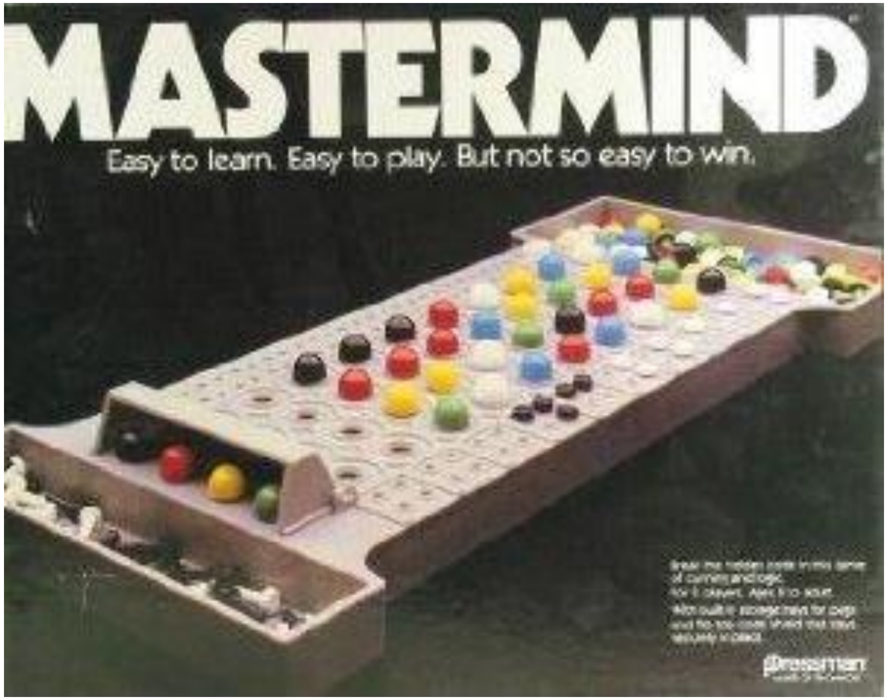
→ quantifies the loss incurred by restricting attention to $\mathcal{B}$

→ identifies essential properties (e.g., learning dependencies between decision variables)

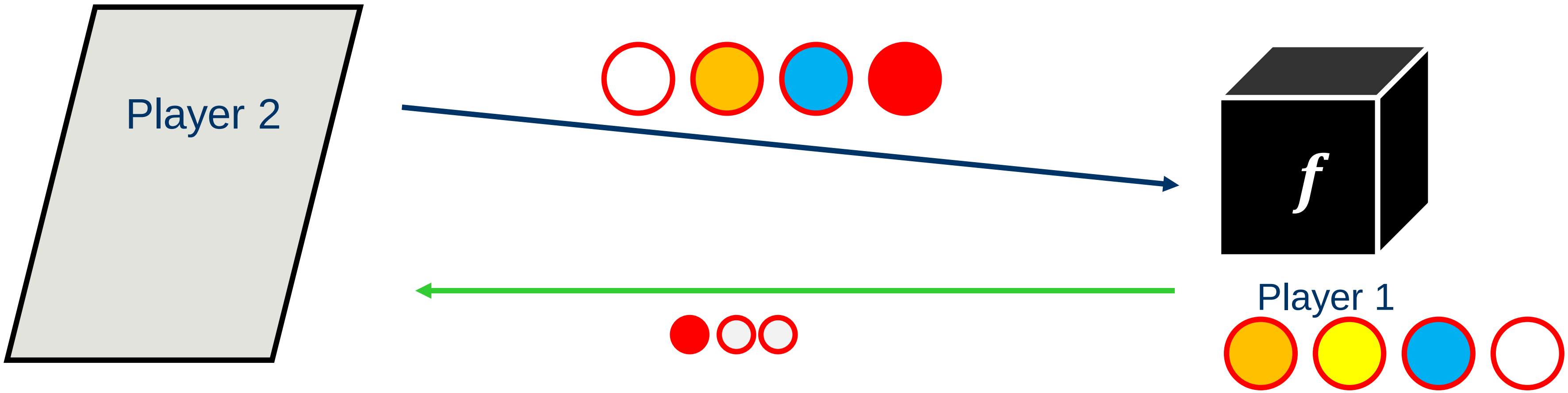
Selected Contributions to Black-Box Complexity Theory:

- Improved bounds for existing models
  - unrestricted model: Mastermind, LeadingOnes
    [Benjamin Doerr, Carola Doerr, Reto Spöhel, Henning Thomas: *Playing Mastermind With Many Colors*. J. ACM 2016]
    [Peyman Afshani, Manindra Agrawal, Benjamin Doerr, Carola Doerr, Kasper Green Larsen, Kurt Mehlhorn: *The query complexity of a permutation-based variant of Mastermind*. Discret. Appl. Math. 2019]
  - unbiased models: tight bounds for 1-ary case, Jump functions in k-ary model
    [Benjamin Doerr, Carola Doerr, Jing Yang: Optimal parameter choices via precise black-box analysis. TCS 2020]
    [Benjamin Doerr, Carola Doerr: The Impact of Random Initialization on the Runtime of Randomized Search Heuristics. Algorithmica 2016]
    [Benjamin Doerr, Carola Doerr, Timo Kötzing: Unbiased Black-Box Complexities of Jump Functions. Evol. Comput. 2015]
  - ranking-based models: tight bound for OneMax
    [Benjamin Doerr, Carola Winzen: Ranking-Based Black-Box Complexity. Algorithmica 2014]
  - memory-restricted models: tight bound for OneMax
    [Benjamin Doerr, Carola Winzen: Playing Mastermind with Constant-Size Memory. Theory Comput. Syst. 2014]
- Design and analysis of new restricted models
  - combinations of restrictions, e.g., memory and unbiased sampling
    [Carola Doerr, Johannes Lengler: *OneMax in Black-Box Models with Several Restrictions*. Algorithmica 2017]
  - elitist black-box model (to quantify potential loss of greedy search)
    [Carola Doerr, Johannes Lengler: *The (1+1) Elitist Black-Box Complexity of LeadingOnes*. Algorithmica 2018. Best paper award at GECCO 2016]
    [Carola Doerr, Johannes Lengler: *Introducing Elitist Black-Box Models: When Does Elitist Behavior Weaken the Performance of Evolutionary Algorithms?* Evol. Comput. 2017]
- Survey and tutorials
  [Carola Doerr: Complexity Theory for Discrete Black-Box Optimization Heuristics. Theory of Evolutionary Computation. Springer 2020]
  [tutorials at GECCO 2013 and 2014, with Benjamin Doerr]

The Mastermind Game

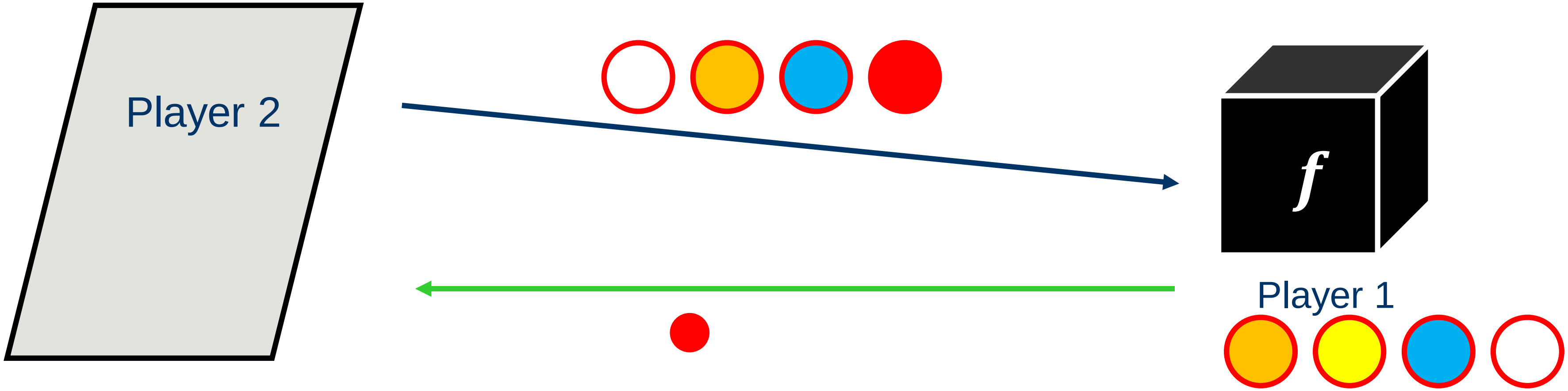


The Mastermind Game



- $f_z : [1..k]^n \to [0..n], x \mapsto (\#\{\, i \mid x_i = z_i \,\},$
$$\max_{\pi} \#\{\, i \ \mid x_{\pi(i)} = z_{\pi(i)} \,\} - \#\{\, i \mid x_i = z_i\})$$

The Mastermind Game



- $f_z\colon [1..k]^n \to [0..n], x \mapsto \#\{\, i \mid x_i = z_i \,\}$
- $\mathcal{F}_k = \{\, f_z \mid z \in [1..k]^n \,\}$
- Theorem [2]: $\mathrm{BBC}(\mathcal{F}_{k=2}) = \Theta(n/\log n)$
- Theorem [3]: $\mathrm{BBC}(\mathcal{F}_{k=n}) = O(n \log n)$
  (several follow-up works improved leading constant and lower order terms)
- Theorem [4]: $\mathrm{BBC}(\mathcal{F}_{k=n}) = O(n \log\log n)$
  - can be achieved in poly-time
  - can be achieved with deterministic algorithms
  - cannot be achieved with non-adaptive algorithms (!)
- Theorem [5]: $\mathrm{BBC}(\mathcal{F}_{k=n}) = \Theta(n)$

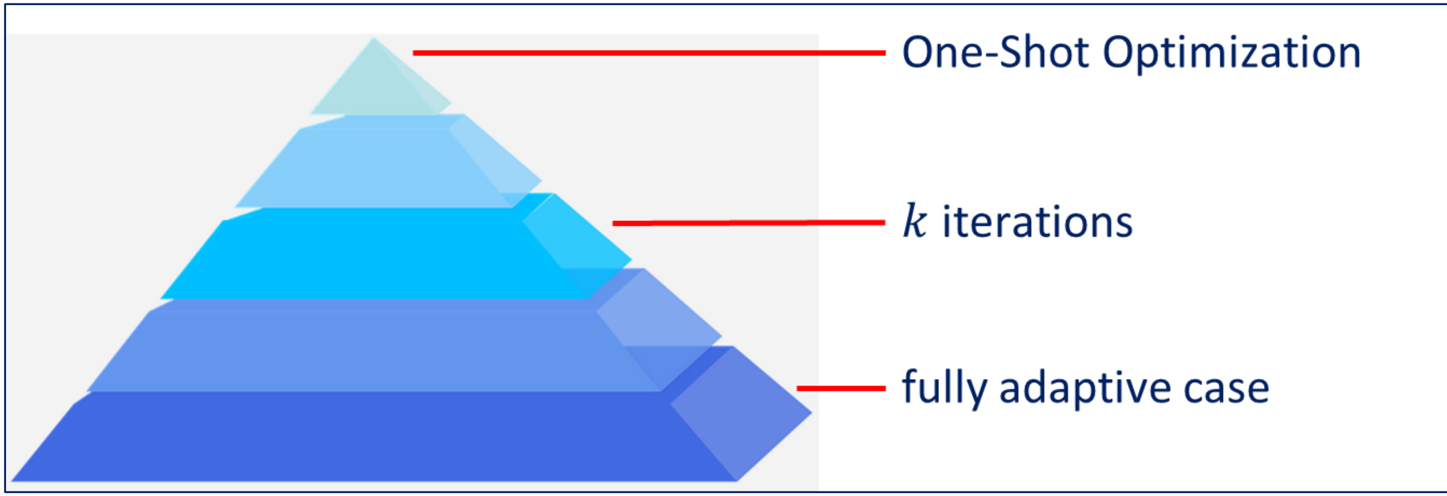
[2] Paul Erdős and Alfréd Rényi. *On two problems of information theory*. Magyar Tudományos Akadémia Matematikai Kutaté Intézet Közleményei, 1963.
[3] Vasek Chvátal: *Mastermind*. Combinatorica 1983
[4] Benjamin Doerr, Carola Doerr, Reto Spöhel, Henning Thomas: *Playing Mastermind With Many Colors*. J. ACM, 2016
[5] Anders Martinsson, Pascal Su: Mastermind with a Linear Number of Queries. CoRR abs/2011.05921 (2020)

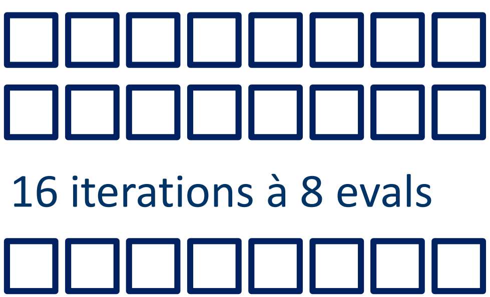
- Theorem [2]: $\mathrm{BBC}(\mathcal{F}_{k=2}) \geq \big(1+o(1)\big)\, n/\log n$

  $\text{1-shot-learning-BBC}(\mathcal{F}_{k=2}) \leq \big(2+o(1)\big)\, n/\log n$

- Theorem [4]: $\mathrm{BBC}(\mathcal{F}_{k=n}) = O(n \operatorname{loglog} n)$

  $\text{1-shot-learning-BBC}(\mathcal{F}_{k=n}) = \Theta(n \log n)$





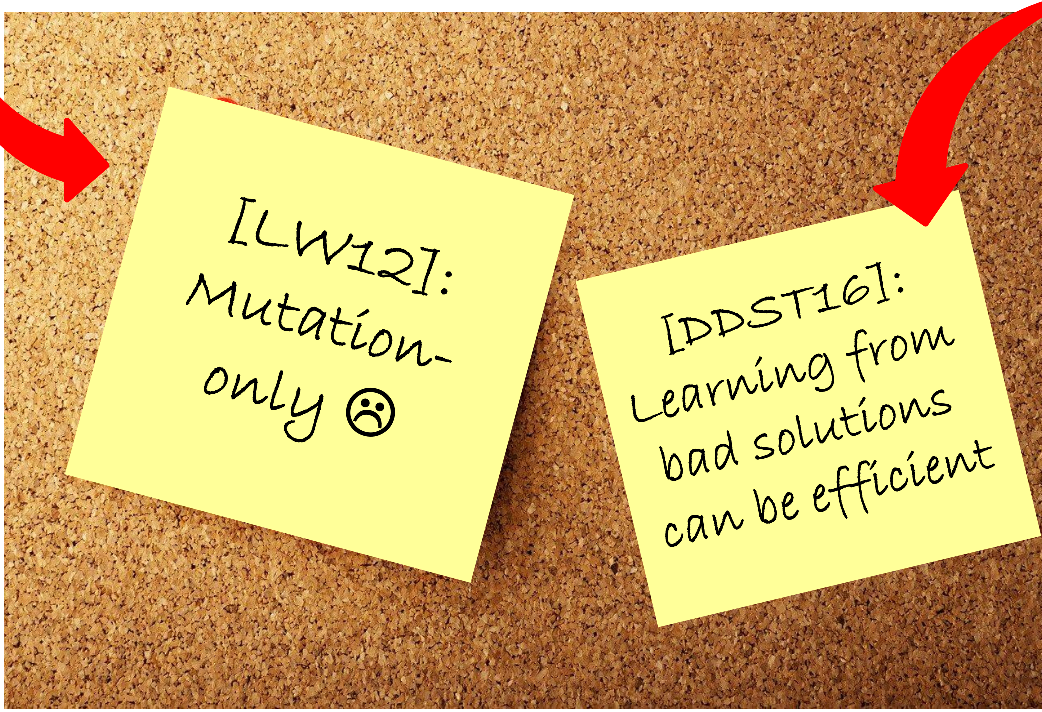
LEADERBOARD

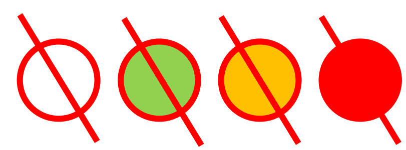
Team	Score	Prize
AutoML.org & IOHprofiler, featuring the switching squirrel	94.845	$3,000
DeepWisdom	93.380	
Huawei Noah's Ark Lab	93.241	
dangnguyen	93.082	

[2] Paul Erdős and Alfréd Rényi. *On two problems of information theory*. Magyar Tudományos Akadémia Matematikai Kutató Intézet Közleményei, 1963.
[4] Benjamin Doerr, Carola Doerr, Reto Spöhel, Henning Thomas: *Playing Mastermind With Many Colors*. J. ACM, 2016

Mastermind with $k = 2$ Colors (OneMax)

- Theorem by Erdős and Rényi: $\mathrm{BBC}(\mathcal{F}_{k=2}) = \Theta(n/\log n)$
- Evolutionary algorithms need $\Omega(n \log n)$
- Lehre, Witt [Algorithmica 2012]:
  The unary unbiased black-box complexity of $\mathcal{F}_{k=2}$ is $\Omega(n \log n)$.
  (that is, all mutation-only algos need at least $n \log n$ evaluations to optimize OneMax)
- How can we use such info to design better algorithms?



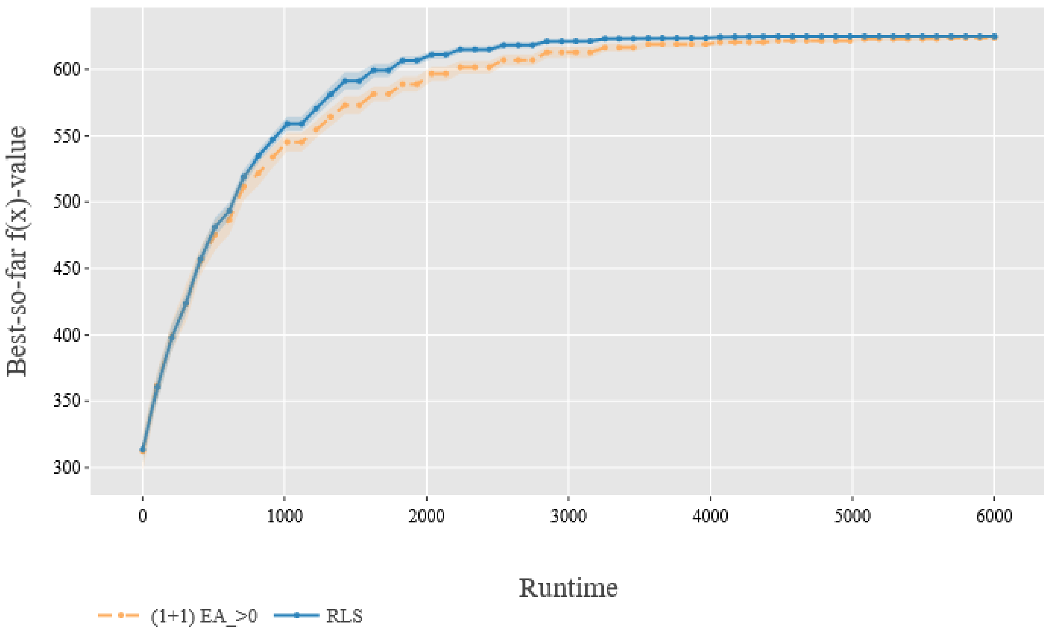


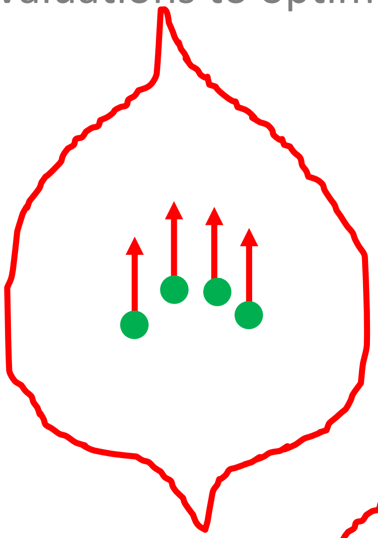
Mastermind with $k = 2$ Colors (OneMax)

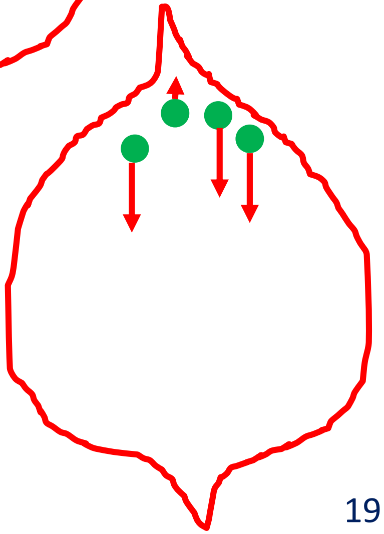
- Theorem by Erdős and Rényi: $\mathrm{BBC}(\mathcal{F}_{k=2}) = \Theta(n/\log n)$
- Evolutionary algorithms need $\Omega(n \log n)$
- Lehre, Witt [Algorithmica 2012]:

  The unary unbiased black-box complexity of $\mathcal{F}_{k=2}$ is $\Omega(n \log n)$.

  (that is, all mutation-only algos need at least $n \log n$ evaluations to optimize OneMax)







Mastermind with $k = 2$ Colors (OneMax)

- Theorem by Erdős and Rényi: $\mathrm{BBC}(\mathcal{F}_{k=2}) = \Theta(n/\log n)$
- Evolutionary algorithms need $\Omega(n \log n)$
- Lehre, Witt [Algorithmica 2012]:

  The unary unbiased black-box complexity of $\mathcal{F}_{k=2}$ is $\Omega(n \log n)$.

  (that is, all mutation-only algos need at least $n \log n$ evaluations to optimize OneMax)

How can we learn from points of inferior objective values?

11001111011111

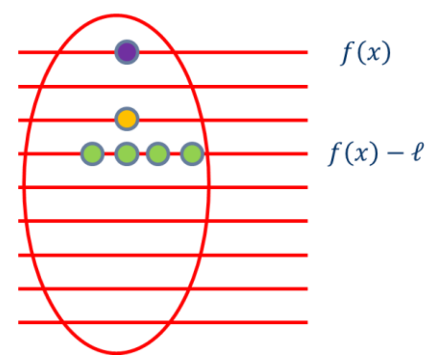
01001011011011

10000110011111

11001111000011

11010111010111

- Assume that we are close to the optimum already
- In this offspring x', at least one bit is correct that is not correct in x



Mastermind with $k = 2$ Colors (OneMax)

- Theorem by Erdős and Rényi: $\mathrm{BBC}(\mathcal{F}_{k=2}) = \Theta(n/\log n)$
- Evolutionary algorithms need $\Omega(n \log n)$
- Lehre, Witt [Algorithmica 2012]:
  The unary unbiased black-box complexity of $\mathcal{F}_{k=2}$ is $\Omega(n \log n)$.
  (that is, all mutation-only algos need at least $n \log n$ evaluations to optimize OneMax)

How can we learn from points of inferior objective values?

$$
\begin{array}{cl}
 & 11001111011111 \\
+ & 11010111010111 \\
\hline
= & 11000111011111 \\
= & 11001111010111 \\
 & \dots \\
= & 11011111011111
\end{array}
$$

- Uniform crossover: take entry from
  - x' with probability c
  - x with probability $1 - c$
- we just need to do this often enough

- Theorem [5]: The $\big(1 + (\lambda, \lambda)\big)$ GA achieves $o(n \log n)$ expected optimization time on 2-color Mastermind.

[5] Benjamin Doerr, Carola Doerr, Franziska Ebel: *From black-box complexity to designing new genetic algorithms*. TCS 2015. Best paper award at GECCO 2013

Mastermind with $k = 2$ Colors (OneMax)

- Theorem by Erdős and Rényi: $\mathrm{BBC}(\mathcal{F}_{k=2}) = \Theta(n/\log n)$
- Evolutionary algorithms need $\Omega(n \log n)$
- Lehre, Witt [Algorithmica 2012]:
  The unary unbiased black-box complexity of $\mathcal{F}_{k=2}$ is $\Omega(n \log n)$.
  (that is, all mutation-only algos need at least $n \log n$ evaluations to optimize OneMax)

How can we learn from points of inferior objective values?

- Uniform crossover: take entry from
  - x' with probability c
  - x with probability $1 - c$
- we just need to do this often enough

- Theorem [4]: The $\left(1 + (\lambda, \lambda)\right)$ GA achieves $o(n \log n)$ expected optimization time on 2-color Mastermind. This proves that ``crossover" is provably better than ``mutation''-only algorithms even for OneMax.

[5] Benjamin Doerr, Carola Doerr, Franziska Ebel: *From black-box complexity to designing new genetic algorithms*. TCS 2015. Best paper award at GECCO 2013

Mastermind with $k = 2$ Colors (OneMax)

- Theorem by Erdős and Rényi: $\mathrm{BBC}(\mathcal{F}_{k=2}) = \Theta(n/\log n)$
- Evolutionary algorithms need $\Omega(n \log n)$
- Lehre, Witt [Algorithmica 2012]:
  The unary unbiased black-box complexity of $\mathcal{F}_{k=2}$ is $\Omega(n \log n)$.
  (that is, all mutation-only algos need at least $n \log n$ evaluations to optimize OneMax)

Theorem [5]: For $k \leq \log n$,
the k-ary unbiased black-box complexity of 2-color Mastermind (OneMax) is $O(n/k)$.

$$
\begin{array}{cc}
 & 11001111011111 \\
+ & 01001111110111 \\
+ & \ldots \\
+ & 11100111011101 \\
+ & 11010101011111 \\
\hline
= & 11111111111111
\end{array}
$$

[5] Benjamin Doerr, Carola Winzen: *Reducing the arity in unbiased black-box complexity*. TCS 2014. Best paper award at GECCO 2012.

- Theorem [4]: The $\big(1 + (\lambda, \lambda)\big)$ GA achieves $o(n \log n)$ expected optimization time on 2-color Mastermind. This proves that ``crossover" is provably better than ``mutation''-only algorithms even for OneMax.

[5] Benjamin Doerr, Carola Doerr, Franziska Ebel: *From black-box complexity to designing new genetic algorithms*. TCS 2015. Best paper award at GECCO 2013

The (1+(λ, λ)) GA

1. **Initialization:** Sample $x \in \{0,1\}^n$ u.a.r.
2. **Optimization: for** $t = 1,2,3, \dots$ **do**
3. **Mutation phase:**
4. Sample ℓ from $\text{Bin}(n, p = \lambda/n)$;
5. **for** $i = 1, \dots, \lambda$ **do** Sample $x^{(i)} \leftarrow \text{mut}_\ell(x)$;
6. Choose $x' \in \{x^{(1)}, \dots, x^{(\lambda)}\}$ with $f(x') = \max\{f(x^{(1)}), \dots, f(x^{(\lambda)})\}$;
7. **Crossover phase:**
8. **for** $i = 1, \dots, \lambda$ **do** Sample $y^{(i)} \leftarrow \text{cross}_{c=1/\lambda}(x, x')$;
9. Choose $y \in \{y^{(1)}, \dots, y^{(\lambda)}\}$ with $f(y) = \max\{f(y^{(1)}), \dots, f(y^{(\lambda)})\}$;
10. **Selection step: if** $f(y) \geq f(x)$ **then** replace x by y;

Adaptive parameter setting works very well:

Theorem [5,6]: For $\lambda = \max\left\{\frac{\mathrm{n}}{\mathrm{n} - f(\mathrm{x})}, 2\right\}$,

the runtime on OneMax is $\Theta(n)$. This is optimal.

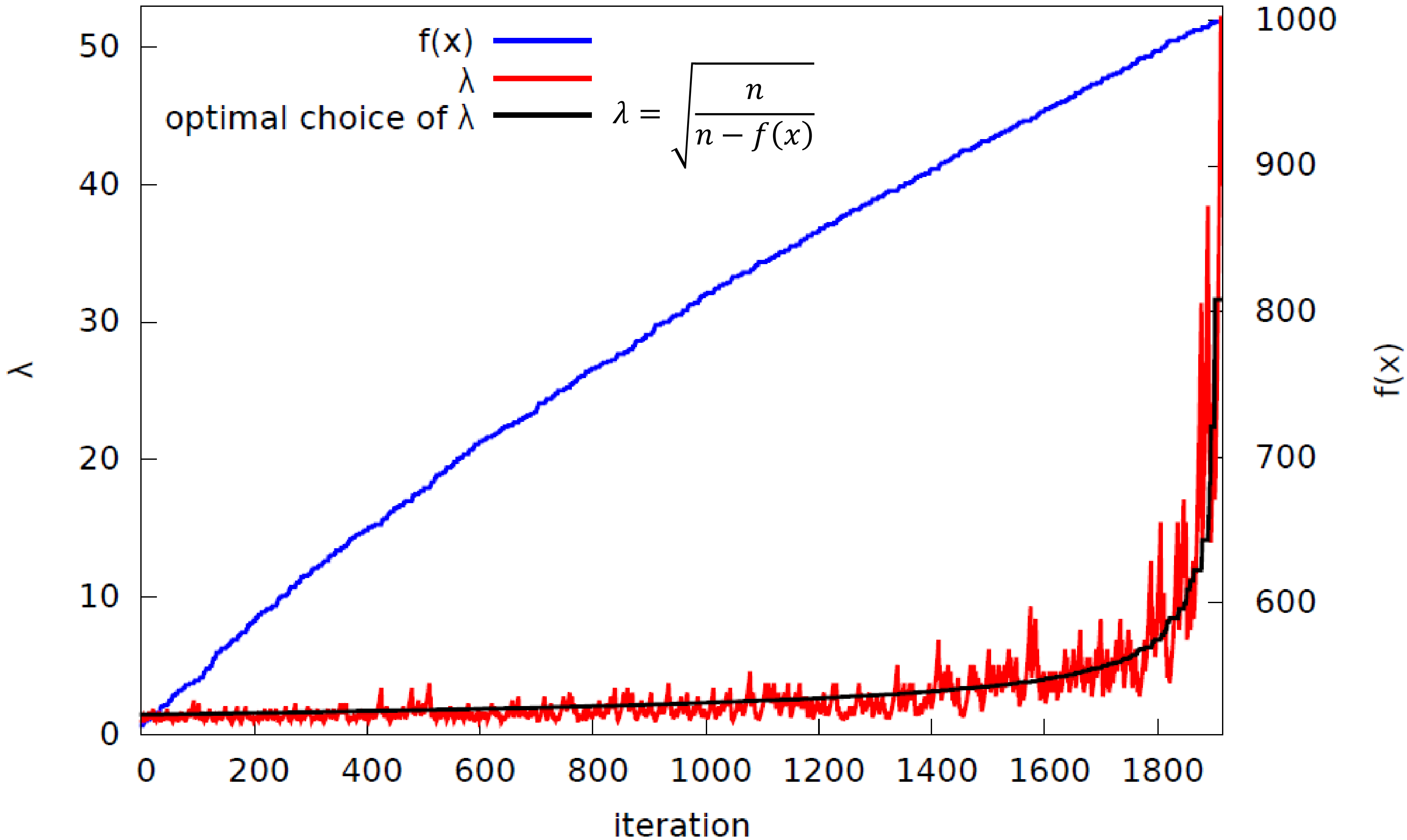
[5] Benjamin Doerr, Carola Doerr, Franziska Ebel: *From black-box complexity to designing new genetic algorithms*. TCS 2015

[6] Benjamin Doerr, Carola Doerr: *Optimal static and self-adjusting parameter choices for the (1+(λ, λ)) genetic algorithm*. Algorithmica 2018

The (1+(λ, λ)) GA

Initialization: Sample $x \in \{0,1\}^n$ u.a.r.
Optimization: for $t = 1,2,3, \ldots$ **do**
  Mutation phase:
    Sample ℓ from $\text{Bin}(n, p = \lambda/n)$;
    for $i = 1, \ldots, \lambda$ **do** Sample $x^{(i)} \leftarrow \text{mut}_\ell(x)$;
    Choose $x' \in \{x^{(1)}, \ldots, x^{(\lambda)}\}$ with $f(x') = \max\{f(x^{(1)}), \ldots, f(x^{(\lambda)})\}$;
  Crossover phase:
    for $i = 1, \ldots, \lambda$ **do** Sample $y^{(i)} \leftarrow \text{cross}_{c=1/\lambda}(x, x')$;
    Choose $y \in \{y^{(1)}, \ldots, y^{(\lambda)}\}$ with $f(y) = \max\{f(y^{(1)}), \ldots, f(y^{(\lambda)})\}$;
  Selection and update step:
    if $f(y) > f(x)$ **then** replace x by y and λ by $F^4\lambda$;
    if $f(y) = f(x)$ **then** replace x by y and λ by λ/F;
    if $f(y) < f(x)$ **then** replace λ by λ/F;

1/5-th success rule
[Rechenberg, Devroye, Schumer/Steiglitz]
here interpretation from:
[Stefan Kern, Sibylle D. Müller, Nikolaus Hansen, Dirk Büche, Jiri Ocenasek, and Petros Koumoutsakos. *Learning probability distributions in continuous evolutionary algorithms - a comparative review*. Natural Computing 2004]



Theorem [6]: The self-adjusting $(1 + (\lambda, \lambda))$ GA with 1/5-th success rule has a linear (and hence optimal) expected running time on OneMax. **No static parameter choice can achieve this, i.e., we have a super-constant speed-up from dynamic parameter choices**

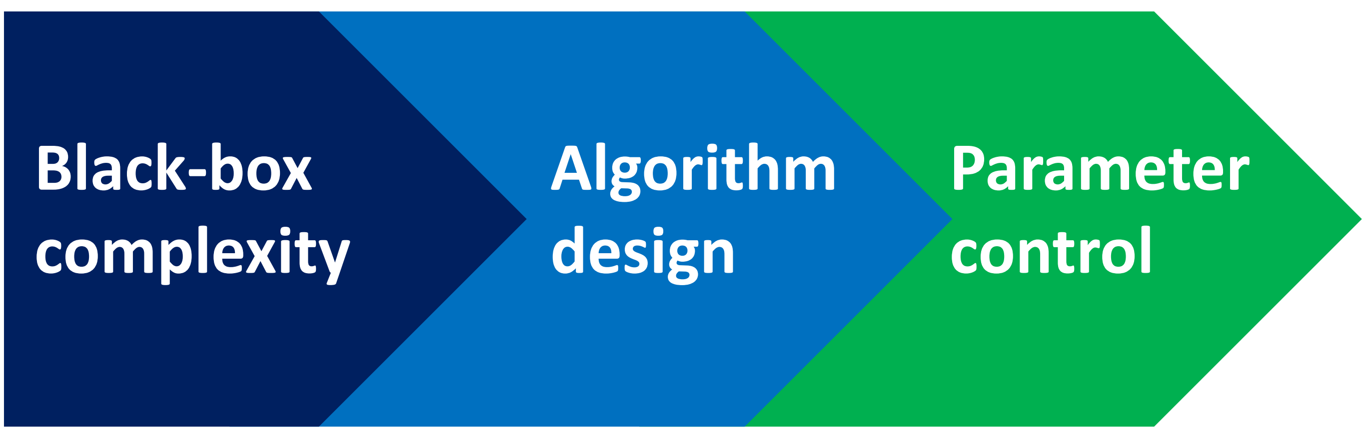
[6] Benjamin Doerr, Carola Doerr: *Optimal static and self-adjusting parameter choices for the (1+(λ, λ)) genetic algorithm*. Algorithmica 2018

Main Contributions

I. Complexity theory for sampling-based heuristics
How do certain algorithmic characteristics influence the performance?

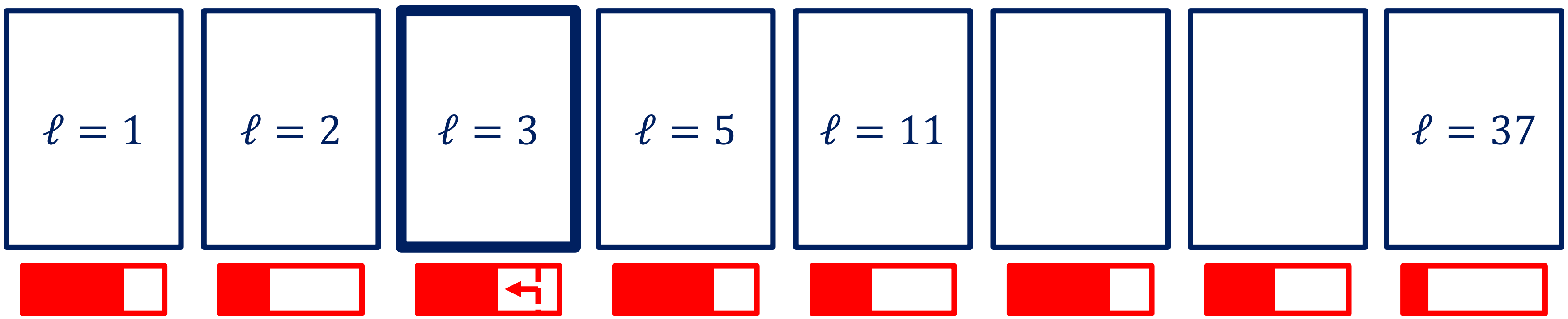
II. Algorithm design
Crossover-based algorithms can be strictly better than mutation-based ones even for OneMax

III. Theoretical analyses for heuristics with dynamic parameter choices
More than just constant-factor speed-up



Other Selected Contributions to Parameter Control

- Theoretical analysis of an ε-greedy RL parameter control technique
  [Benjamin Doerr, Carola Doerr, Jing Yang: *k-Bit Mutation with Self-Adjusting k Outperforms Standard Bit Mutation.* PPSN 2016]



- Key challenge: trade-off between
  - exploitation: we want to maximize reward
  - exploration: has quality of parameter changed?
- MAB-literature: UCB, probability matching, …
- **<u>Our Theorem:</u>** For suitably chosen parameter values, the expected optimization time of the ε-greedy RLS is *almost* optimal: $\mathbb{E}[T] - \mathbb{E}\left[T_{\mathrm{opt},r}\right] = o(n)$.

[11] Álvaro Fialho, Luís Da Costa, Marc Schoenauer, Michèle Sebag: *Analyzing bandit-based adaptive operator selection mechanisms*. Ann. Math. Artif. Intell. 2010

[12] Dirk Thierens: *An adaptive pursuit strategy for allocating operator probabilities*. GECCO 2005

Other Selected Contributions to Parameter Control

- Theoretical analysis of an ε-greedy RL parameter control technique
  [Benjamin Doerr, Carola Doerr, Jing Yang: *k-Bit Mutation with Self-Adjusting k Outperforms Standard Bit Mutation.* PPSN 2016]
- Theoretical analysis for self-adjusting strategy for problems with multi-choice decision variables [Benjamin Doerr, Carola Doerr, Timo Kötzing: *Static and Self-Adjusting Mutation Strengths for Multi-valued Decision Variables*. Algorithmica 2018]
- Lower bounds for algorithms with dynamic parameters
  [Benjamin Doerr, Carola Doerr, Jing Yang: *Optimal parameter choices via precise black-box analysis*. TCS 2020]
  [Benjamin Doerr, Carola Doerr: *Optimal static and self-adjusting parameter choices for the (1+(λ, λ)) genetic algorithm*. Algorithmica 2018]
- Identification of optimal parameter values for RLS, $(1+\lambda)$ EAs, …
  [Nathan Buskulic, Carola Doerr: *Maximizing drift is not optimal for solving OneMax*. Evol. Comput., to appear]
  [Maxim Buzdalov, Carola Doerr: *Optimal Mutation Rates for the (1+λ) EA on OneMax*. PPSN 2020]
- Several empirical results
  [Arina Buzdalova, Carola Doerr, Anna Rodionova: *Hybridizing the 1/5-th Success Rule with Q-Learning for Controlling the Mutation Rate of an Evolutionary Algorithm*. PPSN 2020]
  [Anna Rodionova, Kirill Antonov, Arina Buzdalova, Carola Doerr: *Offspring population size matters when comparing evolutionary algorithms with self-adjusting mutation rates*. GECCO 2019]
  [Furong Ye, Carola Doerr, Thomas Bäck: *Interpolating Local and Global Search by Controlling the Variance of Standard Bit Mutation.* CEC 2019]
  [….]
- Tutorials at GECCO since 2017, WCCI/CEC 2020, PPSN 2018
- Book chapter with survey of theoretical results and new taxonomy
  [Benjamin Doerr, Carola Doerr: *Theory of Parameter Control for Discrete Black-Box Optimization: Provable Performance Gains Through Dynamic Parameter Choices*. Book chapter in Theory of Evolutionary Computation, Springer 2020]

- (1+1) Evolutionary Algorithm with generalized 1/5-th success rule
  - Empirical results in [9]

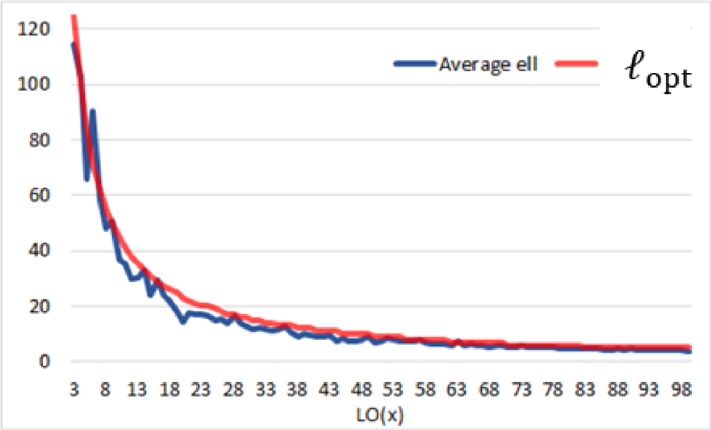
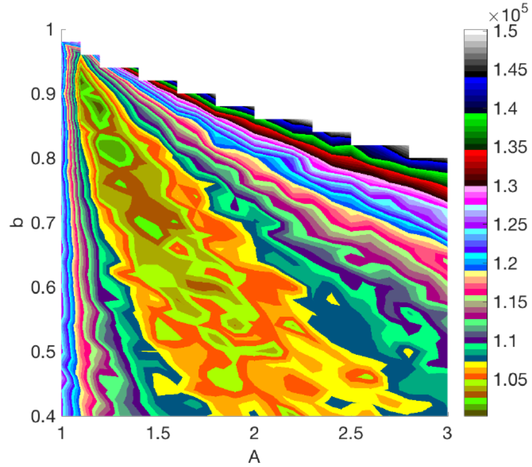


Figure 3: Average and optimal mutation strengths for different Lo(x) values (n = 500, 10 independent runs of the (1 + 1) EA$_\alpha$ with $A = 1.2$, $b = 0.85$, and $p_0 = 1/n$)



  - Theoretical result

    Theorem [10]: The (1+1) EA with $1/e$ success rule achieves asymptotically optimal running time on LeadingOnes.

    (which is around 12% better than that of the best static (1+1) EA)

[9] Carola Doerr, Markus Wagner: *Simple on-the-fly parameter selection mechanisms for two classical discrete black-box optimization benchmark problems.* GECCO 2018

[10] Benjamin Doerr, Carola Doerr, Johannes Lengler: *Self-adjusting mutation rates with provably optimal success rules.* GECCO 2019

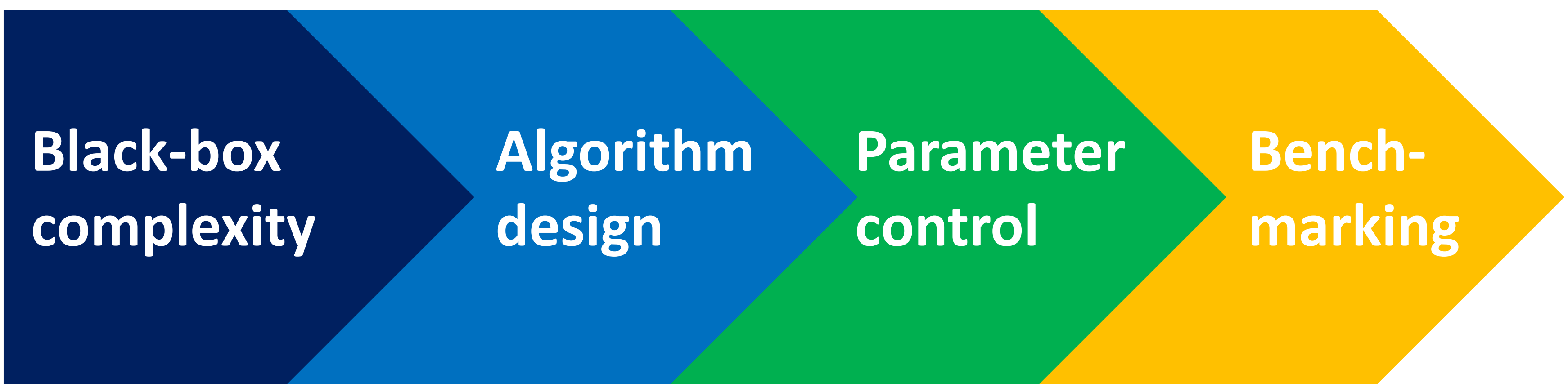
Main Contributions

I. Complexity theory for sampling-based heuristics
How do certain algorithmic characteristics influence the performance?

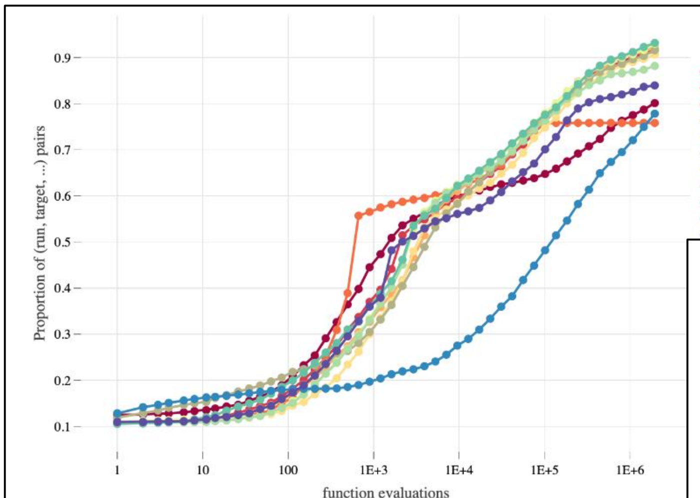
II. Algorithm design
Crossover-based algorithms can be strictly better than mutation-based ones even for OneMax

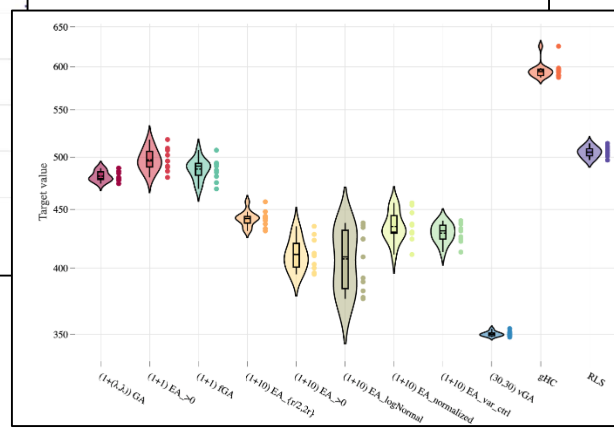
III. Theoretical analyses for heuristics with dynamic parameter choices
More than just constant-factor speed-up

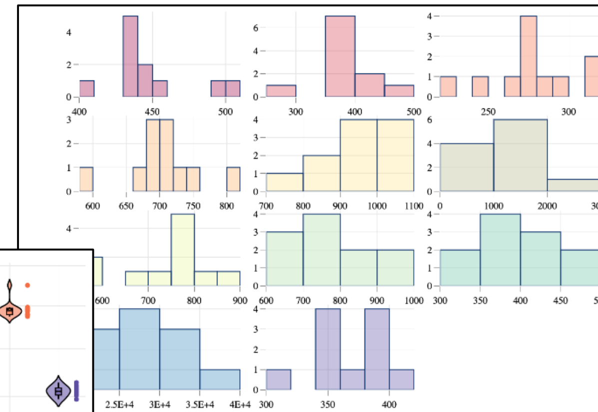
IV. Benchmarking
Modular benchmark design



Benchmarking as Intermediate between Theoretical and Empirical Research

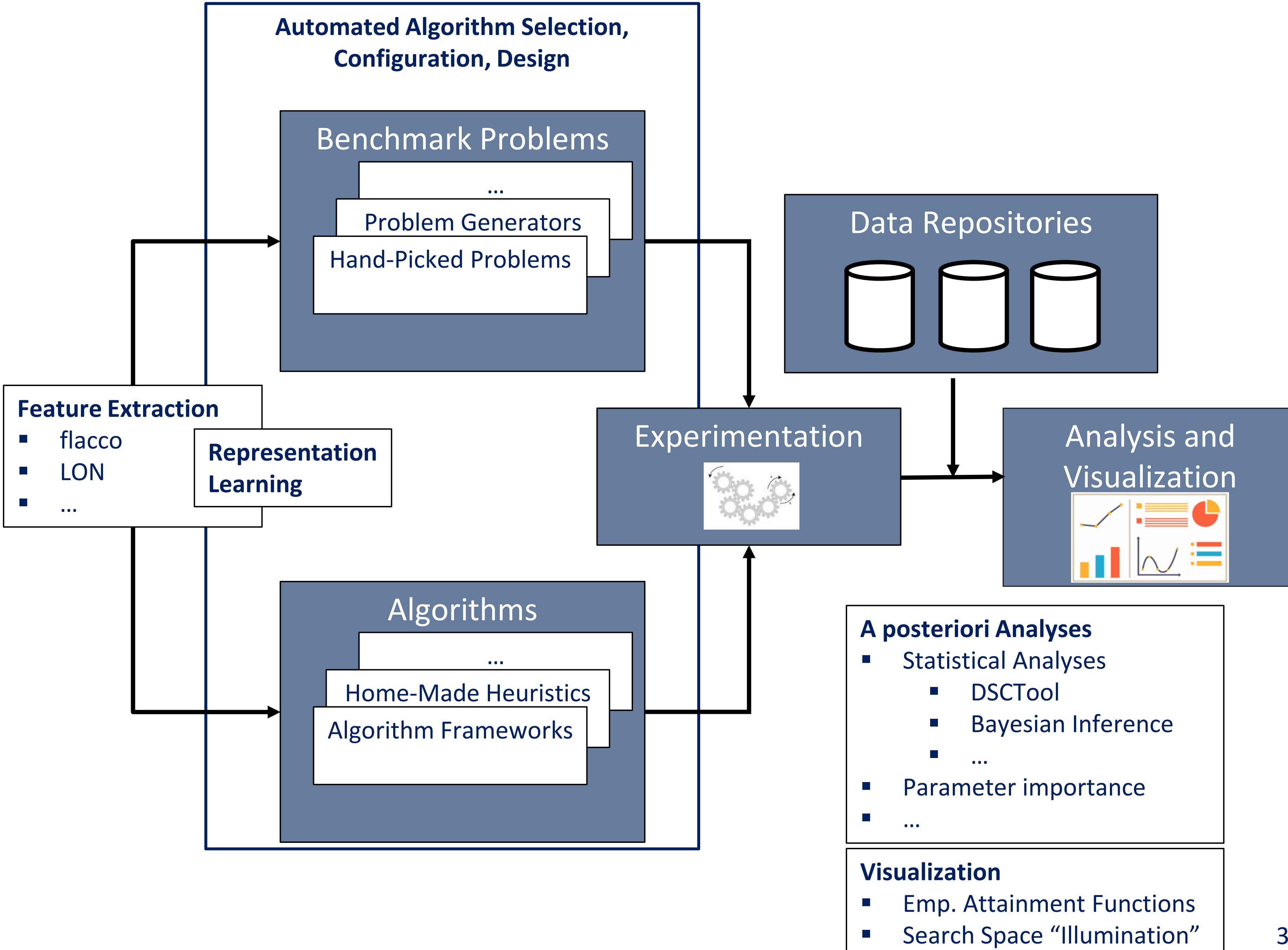


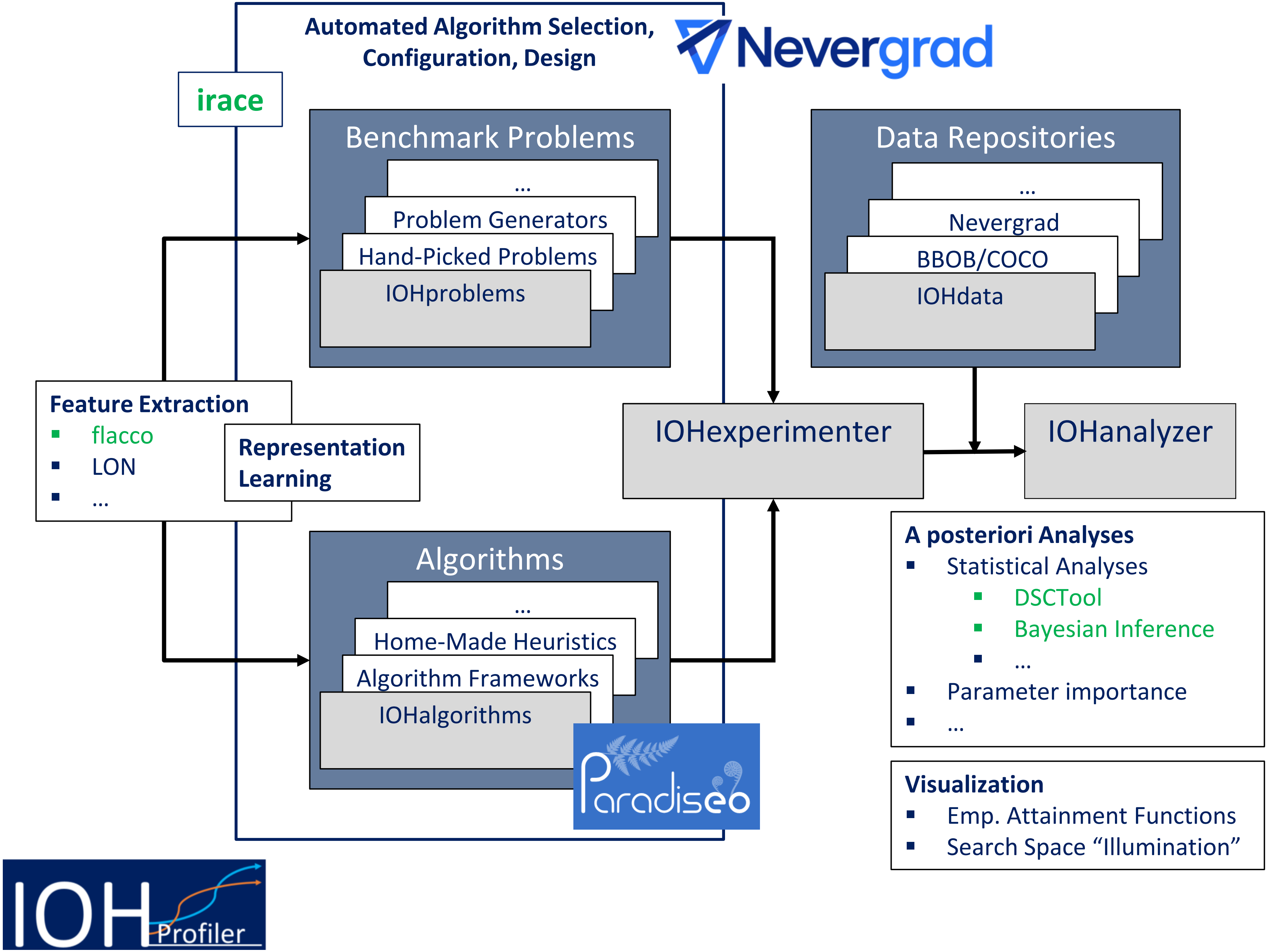


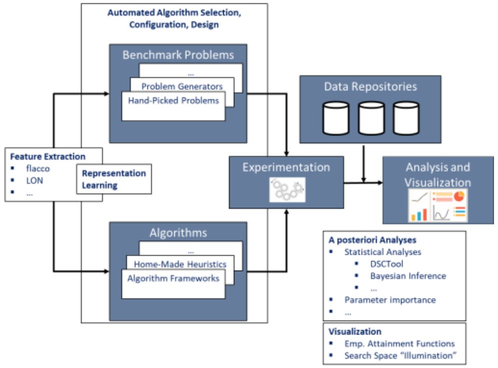


[11] Carola Doerr, Furong Ye, Naama Horesh, Hao Wang, Ofer M. Shir, Thomas Bäck: *Benchmarking discrete optimization heuristics with IOHprofiler*. Appl. Soft Comput. 2020

[12] IOHprofiler is available on GitHub and CRAN. Wiki: https://iohprofiler.github.io/


Automated Algorithm Selection, Configuration, Design
Benchmark Problems
...
Problem Generators
Hand-Picked Problems
Data Repositories
Feature Extraction
flacco
LON
...
Representation Learning
Experimentation
Analysis and Visualization
Algorithms
...
Home-Made Heuristics
Algorithm Frameworks
A posteriori Analyses
Statistical Analyses
DSCTool
Bayesian Inference
...
Parameter importance
...
Visualization
Emp. Attainment Functions
Search Space “Illumination”


Automated Algorithm Selection, Configuration, Design
Nevergrad
irace
Benchmark Problems
...
Problem Generators
Hand-Picked Problems
IOHproblems
Data Repositories
...
Nevergrad
BBOB/COCO
IOHdata
Feature Extraction
flacco
LON
...
Representation Learning
IOHexperimenter
IOHanalyzer
Algorithms
...
Home-Made Heuristics
Algorithm Frameworks
IOHalgorithms
Paradiseo
A posteriori Analyses
Statistical Analyses
DSCTool
Bayesian Inference
...
Parameter importance
...
Visualization
Emp. Attainment Functions
Search Space “Illumination”
IOH Profiler



- ✓ better compatibility between tools
- ✓ better documentation
- ✓ better access to tools, code, data
- ✓ better re-usability (format, ease of access, …)

Lorentz center Online Workshop
Benchmarked
Optimization Meets Machine Learning
9 - 13 November 2020, Leiden, the Netherlands

IEEE TRANSACTIONS ON
EVOLUTIONARY COMPUTATION
A PUBLICATION OF THE IEEE COMPUTATIONAL INTELLIGENCE SOCIETY
www.ieee-cis.org/pubs/tec

Open Optimization Competition 2020

The Nevergrad and IOHprofiler teams are happy to announce that we have paired up for the Open Optimization Competition 2020.

Empirical Approach

- real-world instances
- everything you can implement
- exact numbers
- typically easy to set up
- only a finite number of instances of bounded size
  → representative?
- only tells you numbers
- depends on implementation
- only single algorithms

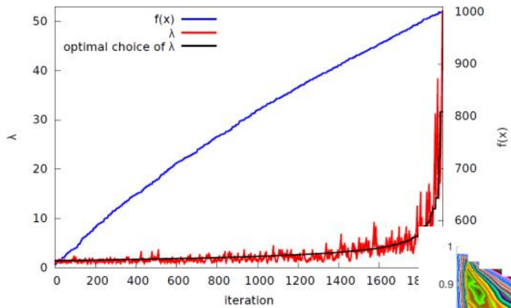
Mathematical Approach

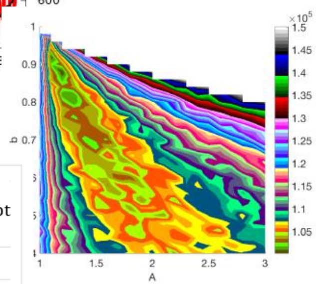
- only models for real-world instances
- limited scope, e.g., (1+1) EA
- limited precision, e.g., $O(n^2)$
- finding proofs can be difficult
- results hold for whole classes of algorithms
  → guarantee!
- proof tells you the reason
- implementation independent
- lower bounds (= performance limits)

Complementary Results

Empirical Approach

Mathematical Approach





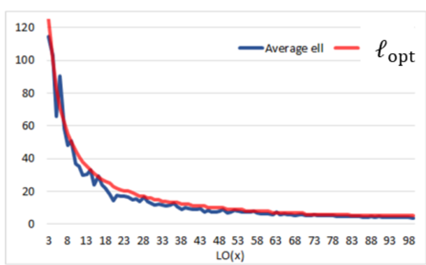


Figure 3: Average and optimal mutation strengths for different Lo(x) values ($n = 500$, 10 independent runs of the $(1+1)$ EA$_\alpha$ with $A = 1.2$, $b = 0.85$, and $p_0 = 1/n$)

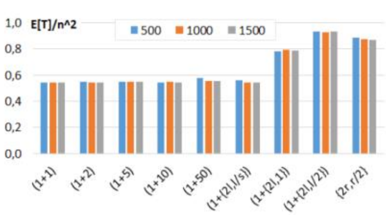
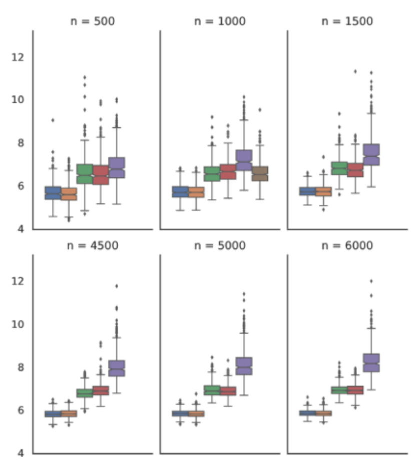


Figure 3: Average optimization times for 100 independent runs, normalized by n^2. (Initial) population size for the adaptive variants is 50.



...

Self-Adjusting Mutation Rates with Provably Optimal Success Rules

Benjamin Doerr, École Polytechnique, CNRS, LIX, Palaiseau, France
Carola Doerr, Sorbonne Université, CNRS, LIP6, Paris, France
Johannes Lengler, ETH, Zürich, Switzerland

Optimal Static and Self-Adjusting Parameter Choices for the $(1+(\lambda,\lambda))$ Genetic Algorithm

Benjamin Doerr[1] · Carola Doerr[2]

Received: 24 October 2016
© Springer Science+Busin

Towards a Theory-Guided Benchmarking Suite for Discrete Black-Box Optimization Heuristics: Profiling $(1+\lambda)$ EA Variants on OneMax and LeadingOnes

Carola Doerr, Sorbonne Université, CNRS, LIP6, Paris, France
Furong Ye, LIACS, Leiden University, Leiden, The Netherlands
Sander van Rijn, LIACS, Leiden University, Leiden, The Netherlands

A Simple Proof for the Usefulness of Crossover in Black-Box Optimization

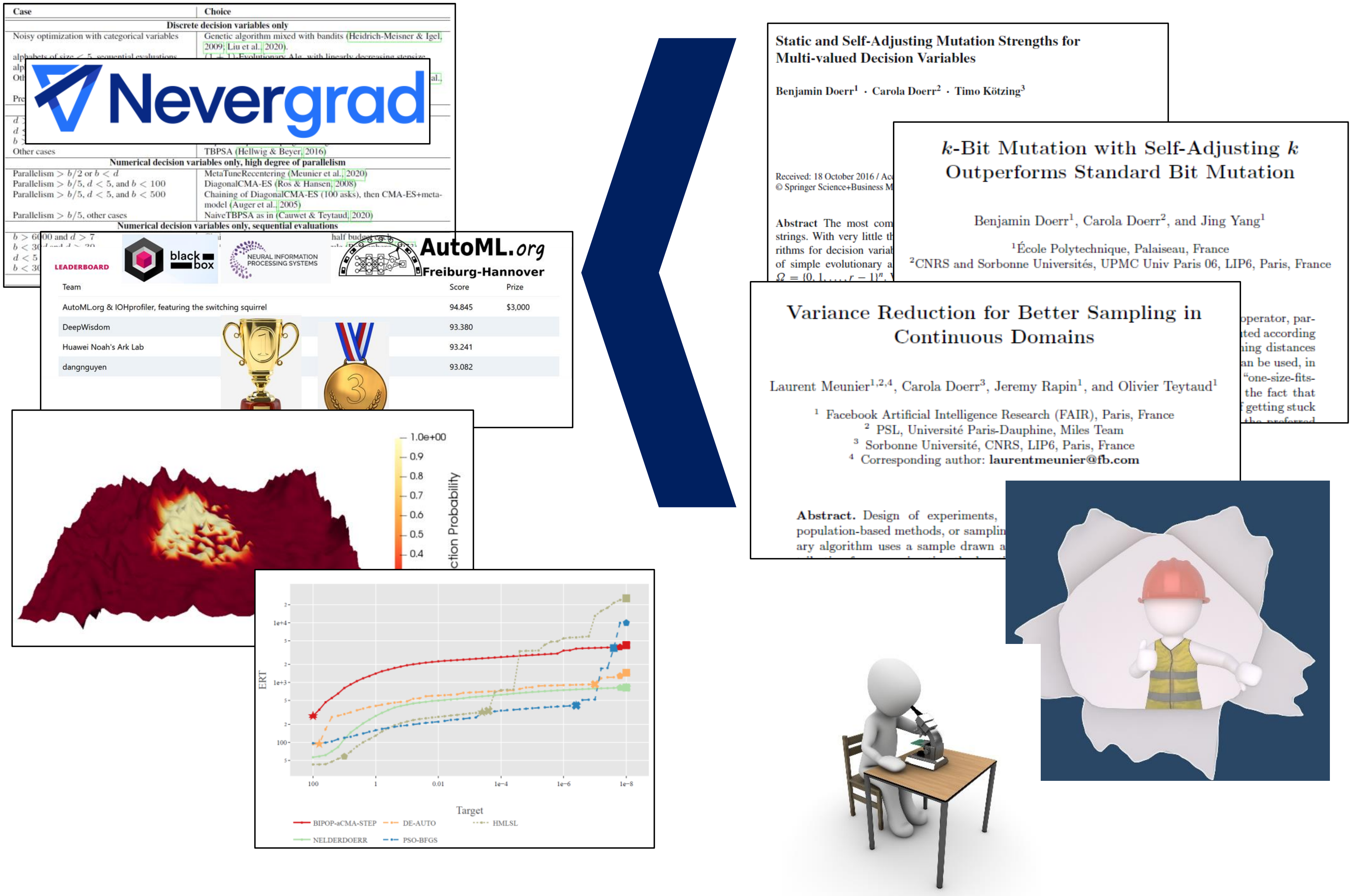
Eduardo Carvalho Pinto[1] and Carola Doerr[2]

[1] DreamQuark, Paris, France
[2] Sorbonne Université, CNRS, LIP6, Paris, France

Abstract. The idea to recombine two or more search points into a new solution is one of the main design principles of evolutionary computation (EC). Its usefulness in the combinatorial optimization context, however, is subject to a highly controversial discussion between EC practitioners and the broader Computer Science research community. While the for-

Empirical Approach

Mathematical Approach



PART II
Academic Activities

Publication List

- 1 software package
- 7 editorials (3 edited proceedings, 2 edited special issues, 2 Dagstuhl reports)
- 4 book chapters
- 30 journal papers
  - 8 Algorithmica
  - 5 Theoretical Computer Science
  - 4 Evolutionary Computation
  - 2 Information Processing Letters
  - 1 Journal of the ACM
  - 1 Theory of Computing Systems
  - 1 Journal of Complexity
  - 1 SIAM Journal on Numerical Analysis
  - 1 Discrete Applied Mathematics
  - 1 Random Structures & Algorithms
  - 1 Distributed Computing
  - 1 ACM Transactions on Economics and Computation
  - 1 Artificial Intelligence
  - 1 Applied Soft Computing
  - 1 Journal of Graph Algorithms and Applications
- 60 conference papers
- 8 tutorials
- 3 theses
- 19 workshop papers and other publications (e.g., articles in lightly refereed conference proceedings and summaries of my work that address a broader scientifically interested audience)

Student Supervision

- **PostDocs**
  - Hao Wang 01/2020-08/2020
    now Assistant Professor at LIACS, Leiden University
  - Martin Krejca 01/2021-10/2022
- **PhD students**
  - Diederick Vermetten (Leiden University, 01/2020-)
    co-supervising with Thomas Bäck
  - Quentin Renau (CIFRE Thales, 02/2019-)
    co-supervising with Benjamin Doerr and Johann Dreo
  - Anja Jankovic (Sorbonne Université, 10/2018-09/2021)
    main supervisor
  - Furong Ye (Leiden University, 10/2017-09/2021)
    co-supervising with Thomas Bäck
  - Jing Yang (École Polytechnique, 10/2015-09/2018)
    co-supervised with Benjamin Doerr
- 14 Master students, 1 Bachelor student, 1 PhD interns

Teaching Activities

- Responsible for the course MPRI 2-24-2 on Solving Optimization Problems with Search Heuristics (with Christoph Dürr)

- Tutorial speaker at ACM GECCO, IEEE WCCI/CEC, PPSN
  - Dynamic parameter choices in evolutionary computation
    - GECCO 2020 and WCCI/CEC 2020 (with Gregor Papa)
    - GECCO 2017, 2018, 2019
    - PPSN 2018
  - Benchmarking and analyzing iterative optimization heuristics with IOHprofiler
    - GECCO 2020 (with Thomas Bäck, Ofer M. Shir, Hao Wang)
    - WCCI/CEC 2020, 2019 (with Thomas Bäck, Ofer M. Shir, Hao Wang)
  - Theory for non-theoreticians
    - GECCO 2016 (with Benjamin Doerr)
    - WCCI/CEC 2016 (with Benjamin Doerr)
  - Black-box complexity: from complexity theory to playing Mastermind
    - GECCO 2014, 2013 (with Benjamin Doerr)

Research Projects/Funding (PI)

- DIM RFSI projects (Paris Ile-de-France region)
  - 2020-22: Optimization Meets Systems Biology (Opt4SysBio)
  - 2019-21: Automated Algorithm Selection for Discrete Black-Box Optimization (AlgoSelect)
  - 2018-20: Online Configuration of Heuristic Optimization Algorithms
- International Emerging Action CNRS/RFBR, with ITMO University, Saint Petersburg, RU
  - 2020-22: Theoretical Foundation of Dynamic Parameter Selection for Randomized Optimization Heuristics
- PGMO projects
  - 2018: Analysis of Evolutionary Algorithms: Beyond Expected Optimization Times (PI)
  - 2017: Self-Adjusting Parameter Choices in Heuristic Optimization (PI)
  - 2016: Parameter Optimization via Drift Analysis (PI)
  - 2014: Towards a Complexity Theory for Black Box Optimization (PI)
- LIP6 laboratory projects
  - 2019: Interactive Multi-objective Optimization (with Thibaut Lust)
- PostDoc Fellowship by the Alexander von Humboldt foundation
- Google Europe PhD Fellowship

Research Projects/Funding (member)

- Vice chair of COST action Improving Applicability of Nature-Inspired Optimisation by Joining Theory and Practice (PI: Thomas Jansen)
- PGMO projects
  - 2020: Understanding and Developing Evolutionary Algorithms via Mathematical Runtime Analyses (member, PI: Benjamin Doerr)
  - 2019: Passive Radar Coverage Optimization (member, PI: Benjamin Doerr)
  - 2015: How Randomness Helps in Scheduling Problems (member, PI: Fanny Pascual)

Selected Community Services

- Editorial Activities:
  - Associate Editor: ACM Transactions on Evolutionary Learning and Optimization
  - Editorial Board member: Evolutionary Computation Journal
  - Guest editor for a special issue in IEEE Transactions on Evolutionary Computation on *Benchmarking Sampling-Based Optimization Heuristics: Methodology and Software* (BENCH), with Thomas Bäck, Bernhard Sendhoff, and Thomas Stützle
  - Guest editor for two special issues in Algorithmica:
    - 2017, with Francisco Chicano (for GECCO theory track 2015)
    - 2019, with Dirk Sudholt (for GECCO theory track 2017)
  - Review editor: Optimization (Frontiers in Applied Mathematics and Statistics)
  - Advisory board: Springer Natural Computing Book Series
- Program Committee Chair
  - PPSN 2020 (268 submissions, 99 accepted)
  - ACM FOGA (31 submissions, 15 accepted)
  - ACM GECCO theory track 2017 and 2015
- EC Technical Committee of the IEEE Computational Intelligence Society (member since 2020)
- Board member of the GT CoA of GDR-IM [French Algorithms and Complexity group]
- Conseil scientifique de l'UFR (≈ scientific board of engineering department at Sorbonne U.)

Event Organization

- Benchmarking Network: consolidate and stimulate activities on benchmarking iterative optimization heuristics, https://sites.google.com/view/benchmarking-network/
- Workshops
  - 2 Dagstuhl seminars on Theory of Randomized Optimization Heuristics (‘17, ‘19)
  - Lorentz Center Workshop Benchmarked: Optimization meets Machine Learning
  - several workshops on benchmarking@GECCO, PPSN, CEC, women@GECCO, …
- Competitions
  - Open Optimization Competition (joint effort of IOHprofiler and Nevergrad team@Facebook) 2021, 2020
- Special Session
  - Representation Learning for Meta-Heuristic Optimization at CEC 2021
- Summer School
  - COST action Summer School on Theory and Applications of Nature-Inspired Optimization Heuristics (2017)
- Other activities
  - Hot off the Press Chair at GECCO 2021
  - Late Breaking Abstracts chair at GECCO 2019
  - Tutorials chair at PPSN 2016 (with Nicolas Bredeche)