Exploratory landscape analysis

Mateusz Zaborski
M.Zaborski@mini.pw.edu.pl

Faculty of Mathematics and Information Science
Warsaw University of Technology

17.11.2021

Table of Contest

Algorithm selection problem

No-free-lunch (NFL) theorem [Wolpert et al., 1995]

"(...) all algorithms that search for an extremum of a cost function perform exactly the same, according to any performance measure, when averaged over all possible cost functions"

Performance complementarity [Kerschke et al., 2019]

"(...) different algorithms perform best on different types of problem instances"

Observed (among others) for:

- planning and scheduling problems
- mixed integer programming
- propositional satisfiability (SAT)
- constraint satisfaction
- travelling salesperson problem
- machine learning
- polynomial-time-solvable problems
- **continuous optimisation**

Per-instance algorithm selection

How to determine **a priori** which algorithm should be used to solve a given instance?

Per-instance algorithm selection problem - given a computational problem, a set of algorithms for solving this problem, and a specific instance that needs to be solved, determine which of the algorithms can be expected to perform best on that instance [Rice, 1976].

Per-instance algorithm selection

Definition 1 (Per-instance algorithm selection problem). *Given*

- *a set $\mathcal{I}$ of problem instances drawn from a distribution $\mathcal{D}$,*
- *a space of algorithms $\mathcal{A}$, and*
- *a performance measure $m : \mathcal{I} \times \mathcal{A} \to \mathbb{R}$,*

the per-instance algorithm selection problem is to find a mapping $s : \mathcal{I} \to \mathcal{A}$ that optimizes $\mathbb{E}_{i \sim \mathcal{D}} m(i, s(i))$, i.e., the expected performance measure for instances i distributed according to $\mathcal{D}$, achieved by running the selected algorithm $s(i)$ for instance i.

Figure 1: Source: [Bischl et al., 2016]

Per-instance algorithm selection

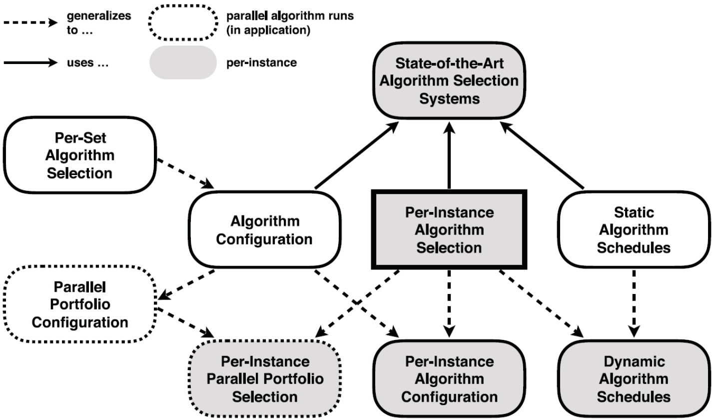


Figure 2: Connections between per-instance algorithm selection and related problems [Kerschke et al., 2019]

Single-objective continuous optimization

Single-objective continuous black box optimization problem

- Goal - minimize an objective function (or fitness function or cost function)

$$f : \mathbb{R}^n \rightarrow \mathbb{R}$$

- Single-objective continuous optimization
- Black Box scenario
  - Function values of evaluated search points are the only accessible information
  - Gradients are not available
- Search cost - number of function evaluations

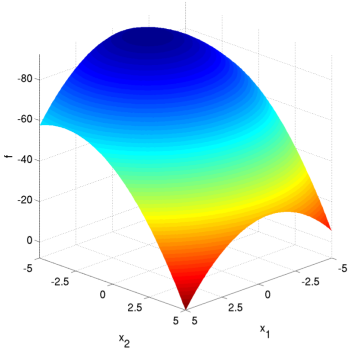


Figure 3: Sphere 2D function from COCO BBOB benchmark [Hansen et al., 2019]

Single-objective continuous black box optimization problem

Problems

Problems:

- Infinite number of solutions in a continuous domain
- Multidimensional problems are difficult for grid search
- The goal is to find a global (not local) optimum
- Unknown function shape
  - Non-linear, non-quadratic
  - Discontinuities, sharp ridges
  - Non-separability
  - Ill-conditioning

Single-objective continuous black box optimization problem

Examples

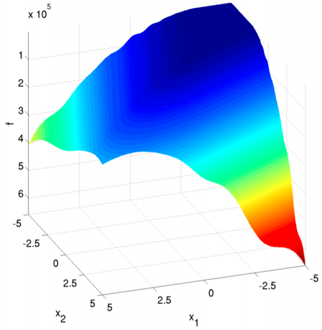


Figure 4: Attractive Sector Function (2D)[Hansen et al., 2019]

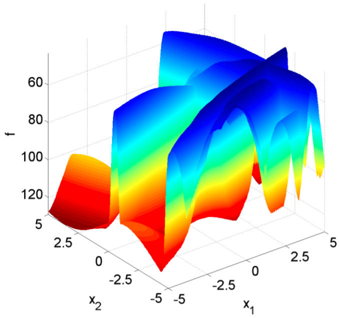


Figure 5: Gallagher's Gaussian 21-hi Peaks Function (2D)[Hansen et al., 2019]

Single-objective continuous black box optimization problem

Examples

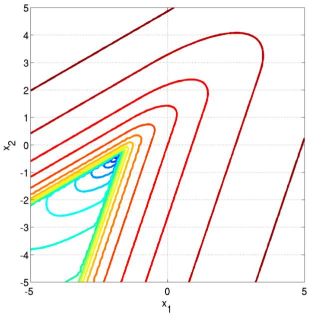


Figure 6: Attractive Sector Function (2D)[Hansen et al., 2019]

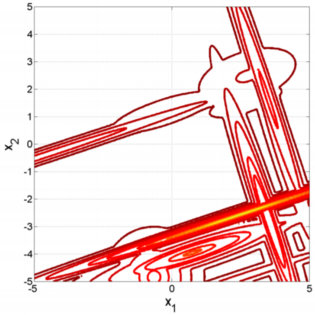


Figure 7: Gallagher's Gaussian 21-hi Peaks Function (2D)[Hansen et al., 2019]

Exploratory landscape analysis

Exploaratory landscape analysis

Initial work

Exploratory Landscape Analysis

Olaf Mersmann *
TU Dortmund, Germany
olafm@statistik.tu-dortmund.de

Bernd Bischl
TU Dortmund, Germany
bischl@statistik.tu-dortmund.de

Heike Trautmann
TU Dortmund, Germany
trautmann@statistik.tu-dortmund.de

Mike Preuss
TU Dortmund, Germany
mike.preuss@tu-dortmund.de

Claus Weihs
TU Dortmund, Germany
weihs@statistik.tu-dortmund.de

Günter Rudolph
TU Dortmund, Germany
guenter.rudolph@tu-dortmund.de

Figure 8: ELA initial work (2011) [Mersmann et al., 2011]

Exploaratory landscape analysis

Motivation [Mersmann et al., 2011]

- Once a problem is well known, one can employ a matching optimization algorithm to solve it
- Most problems encountered in practice (e.g. from the engineering domain) are poorly understood
- If computing one fitness evaluation is costly, initial testing or parameter tuning are problematic
- Finding interactions between problem properties and algorithms is crucial.
- Problem features defining a specific algorithm's performance can be gathered without actually running it.
- Properties are estimated using a small sample of function values combined with statistical and machine learning techniques.

Exploaratory landscape analysis

Principle

Principle:

1. 6 low-level feature sets introduced (fitness landscape characterization)
   - curvature
   - convexity
   - levelset
   - local search
   - meta models
   - y-distribution
2. 50 sub-features
3. Latin Hypercube Sampling ($D^s = [X^s, Y^s]$)
4. Prediction of BBOB fetures / high-level features

COCO BBOB functions

COCO BBOB functions groups [Hansen et al., 2019]:

- Separable functions: f1 - f5
- Functions with low or moderate conditioning: f6 - f9
- Functions with high conditioning and unimodal: f10 - f14
- Multimodal functions with adequate global structure: f15 - f19
- Multimodal functions with weak global structure: f20 - f24

Exploaratory landscape analysis

High-level features

Multi-modality refers to the number of local optima of a problem. In practical applications, many problems are not unimodal (convex) as favoured by most classical optimization algorithms.

Global structure is what remains after deleting all non-optimal points. For Rastrigins problem, we obtain a perforated parabola which is unimodal. Problems without global basin structure are more difficult because one virtually needs to look in every corner.

Separability means a problem may be partitioned into subproblems which are then of lower dimensionality and should be considerably easier to solve. However, for an unknown problem, information about its separability may be scarce.

Variable scaling can make a problem behave differently in each dimension. It can be essential to perform small steps in some dimensions, and large ones in others, which is due to the non-spherical form of basins of attraction. Note that scaling may differ between different basins of attraction.

Figure 9: High-level features [Mersmann et al., 2010]

Exploaratory landscape analysis

High-level features

Search space homogeneity refers to a search space without phase transitions. Its overall appearance is similar in different search space areas. Most benchmark problems are of this type.

Basin size homogeneity means the size relation (largest to smallest) of all basins of attraction (e.g. [12] postulated that size differences influence problem hardness).

Global to local optima contrast refers to the difference between global and local peaks in comparison to the average fitness level of a problem. It thus determines if very good peaks are easily recognized as such.

Plateaus can make the life of optimization algorithms a lot harder as they do not provide any information about good directions to turn to. However, in the BBOB'09 test set, this property is largely unused.

Figure 10: High-level features [Mersmann et al., 2010]

Exploaratory landscape analysis

Low-level features [Mersmann et al., 2010]

Feature group and name	Description
Meta-model features:	
1 approx.{linear,lineari}_ar2	adjusted R^2 of the estimated linear regression model without and with interaction
1 approx.linear_{min,max}_coef	minimum and maximum value of the absolute values of the linear model coefficients
2 approx.{quadratic,quadratici}_ar2	adjusted R^2 of the estimated quadratic regression model without or with interaction
2 approx.quadratic_cond	maximum absolute value divided by minimum absolute value of the coefficients of the quadratic terms in the quadratic model
Convexity features:	
3 convex.{linear,convex}_p	estimated probability of linearity and convexity
3 convex.linear_dev	mean deviation from linearity
y distribution features:	
4 distr.skewness_y	skewness of the distribution of the function values
4 distr.kurtosis_y	kurtosis of the distribution of the function values
4 distr.n_peaks	estimation of the number of peaks in the distribution of the function values
Levelset features:	
5 levelset.lda_mmce_{10,25,50}	mean LDA misclassification error for function values split by 0.1, 0.25, 0.5 quantile (estimated by CV)
5 levelset.lda_vs_qda_{10,25,50}	levelset.lda_mmce_{10,25,50} divided by levelset.qda_mmce_{10,25,50}
6 levelset.qda_mmce_{10,25,50}	mean QDA misclassification error for function values split by 0.1, 0.25, 0.5 quantile (estimated by CV)
7 levelset.mda_mmce_{10,25,50}	mean MDA misclassification error for function values split by 0.1, 0.25, 0.5 quantile (estimated by CV)
Local search features:	
8 ls.n_local_optima	number of local optima estimated by the number of identified clusters
8 ls.best_to_mean_contrast	minimum value of cluster centers divided by the mean value of cluster centers
8 ls.{best,worst}_basin_size	proportion of points in the best and worst cluster
8 ls.mean_other_basin_size	mean proportion of points in all clusters but the cluster with the best cluster center
8 ls.{min,lq,med,uq,max}_feval	0, 0.25, 0.5, 0.75 and 1 quantile of the distribution of the number of function evaluations performed during a single local search
Curvature features:	
9 numderiv.grad_norm_{min,lq,med,uq,max}	minimum, lower quantile, median, upper quantile and maximum of the euclidean norm of the estimated numerical gradient
9 numderiv.grad_scale_{min,lq,med,uq,max}	minimum, lower quantile, median, upper quantile and maximum of the maximum divided by the minimum of the absolute values of the estimated partial gradients
10 numderiv.hessian_cond_{min,lq,med,uq,max}	minimum, lower quantile, median, upper quantile and maximum of the maximum divided by the minimum eigenvalue of the estimated hessian matrix

COCO BBOB functions

Function	multim.	gl.-struc.	separ.	scaling	homog.	basins	gl.-loc.
1 Sphere	none	none	high	none	high	none	none
2 Ellipsoidal separable	none	none	high	high	high	none	none
3 Rastrigin separable	high	strong	none	low	high	low	low
4 Bueche-Rastrigin	high	strong	high	low	high	med.	low
5 Linear Slope	none	none	high	none	high	none	none
6 Attractive Sector	none	none	high	low	med.	none	none
7 Step Ellipsoidal	none	none	high	low	high	none	none
8 Rosenbrock	low	none	none	none	med.	low	low
9 Rosenbrock rotated	low	none	none	none	med.	low	low
10 Ellipsoidal high-cond.	none	none	none	high	high	none	none
11 Discus	none	none	none	high	high	none	none
12 Bent Cigar	none	none	none	high	high	none	none
13 Sharp Ridge	none	none	none	low	med.	none	none
14 Different Powers	none	none	none	low	med.	none	none
15 Rastrigin multi-modal	high	strong	none	low	high	low	low
16 Weierstrass	high	med.	none	med.	high	med.	low
17 Schaffer F7	high	med.	none	low	med.	med.	high
18 Schaffer F7 mod. ill-cond.	high	med.	none	high	med.	med.	high
19 Griewank-Rosenbrock	high	strong	none	none	high	low	low
20 Schwefel	med.	deceptive	none	none	high	low	low
21 Gallagher 101 Peaks	med.	none	none	med.	high	med.	low
22 Gallagher 21 Peaks	low	none	none	med.	high	med.	med.
23 Katsuura	high	none	none	none	high	low	low
24 Lunacek bi-Rastrigin	high	weak	none	low	high	low	low

Figure 11: Classification of the noiseless BBOB functions based on their properties (multi-modality, global structure, separability, variable scaling, homogeneity, basin-sizes, global to local contrast). Predefined groups are separated by horizontal lines. [Mersmann et al., 2011]

Exploaratory landscape analysis

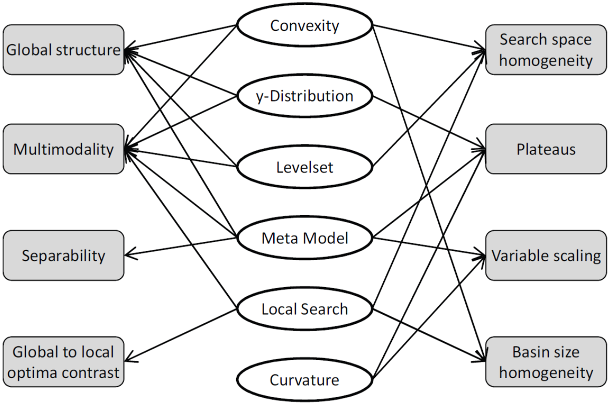


Figure 12: Relationships between high-level features (grey) and low-level feature classes (white) [Mersmann et al., 2011]

Exploaratory landscape analysis

Feature Group	**Cost of selected bits**
convex	$\sum_{i=1}^{l} \tilde{s}_i + l \cdot 1000$
ls	$\sum_{i=1}^{l} (\tilde{s}_i + 50 \cdot d \cdot ls_{\bar{FE}}(s_i))$
numderiv.grad	$\sum_{i=1}^{l} \tilde{s}_i + l \cdot 100 \cdot d^2$
numderiv.hessian	$\sum_{i=1}^{l} \tilde{s}_i + l \cdot 100 \cdot d^3$
all others	$\sum_{i=1}^{l} \tilde{s}_i$

Figure 13: Cost (i.e. NFE) of the selected bits. The vector $\widetilde{s} = (\widetilde{s}_1, \ldots, \widetilde{s}_l)$contains the different initial design sizes within the selected features; ls_{FE} reflects the mean number of FE of all local searches for the respective initial design; d equals the dimensionality of the test function. [Mersmann et al., 2011]

Problem 1

Feature selection for BBOB groups

- Random Forest (RF) classification algorithm
- 5-fold cross-validation
- Target: 5 predefined BBOB groups based
- 24 x 5 x 5 x 5 = 3000 function instances (# functions x # instances x # dimensions x # repetitions)
- $s = \{5000, 2500, 1250, 625, 500, 400, 300, 200, 100\}$
- Multi-objective optimization using SMS-GA (based on SMS-EMOA)
  - minimize misclassification rate (MCE)
  - minimize cost of calculating active features (NSF)
  - minimize total number of feature groups used (NFE)

Problem 2

Feature selection for high-level features

- Problem 1 extension
- MCE is computed for each of 7 classification problems
- maximum of these errors (MaxMCE) is used

Results

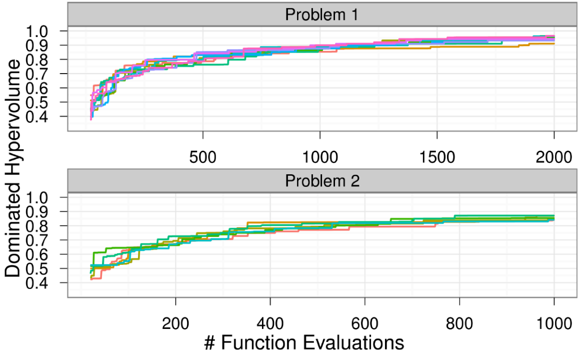


Figure 14: Plot of the hypervolume dominated by the active population after each function evaluation for the two optimization problems (10D). The colors denote the different runs of the SMSGA. [Mersmann et al., 2011]

Results

Problem 1

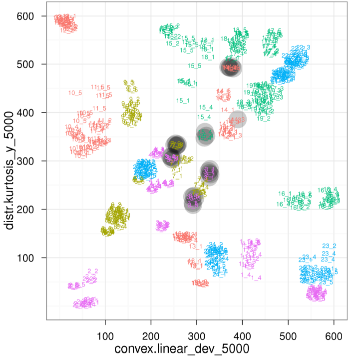
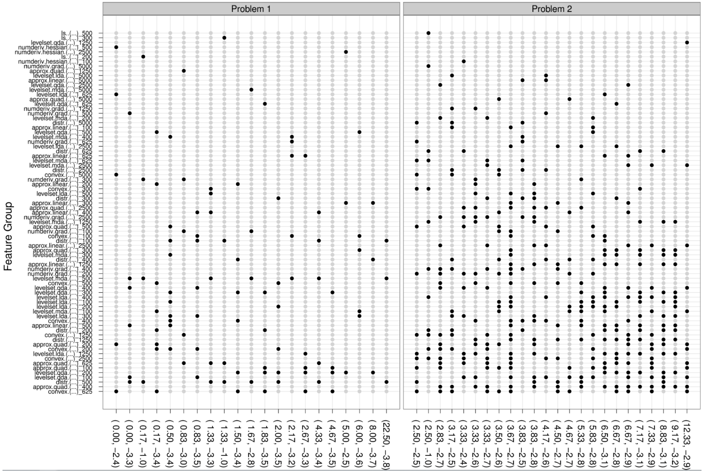


Figure 15: Scatter plot of two features, selected via forward selection and cross-validation on the function instances (10D). [Mersmann et al., 2011]

Results [Mersmann et al., 2011]



Problem *leave-onefunction-out problem*

- Preliminary work in this direction
- 4-class problem: accuracy 73%
- 2-class problem: accuracy $> 96\%$

Further research

Further research

Ref.	Year	Task	Dimension n
[25]	2011	HP	$n \in \{5, 10, 20\}$
[32]	2012	PP	$n \in \{2, 3, 5, 10, 20\}$
[5]	2012	AS	$n \in \{5, 10, 20\}$
[16]	2014	HP	$n = 2$
[17]	2015	HP	$n \in \{2, 3, 5, 10\}$
[33]	2015	HP	$n \in \{2, 5, 10, 20\}$
[18]	2016	HP	$n \in \{2, 3, 5, 10\}$
[2]	2016	PIAC	$n \in \{2, 3, 5, 10\}$
[4]	2017	PIAC	train: $n \in \{2, 4, 5, 8, 10, 16, 20, 32, 40, 64\}$ test: $n \in \{2, 4, 8, 10, 16, 20, 32, 40, 50, 66, 100\}$
[7]	2019	HP, AS	$n \in \{2, 3, 5\}$
[19]	2019	AS	$n \in \{2, 3, 5, 10\}$
[13]	2020	PP	$n = 5$
[39]	2020	HP	$n = 5$
[8]	2020	HP	$n = 5$

Figure 16: Dimension n of the BBOB functions in selected previous studies for the following four tasks: high-level property classification (HP), algorithm selection (AS), performance prediction (PP), and per-instance algorithm configuration (PIAC). [Tanabe, 2021]

Automated Algorithm Selection - ELA + ML [Kerschke and Trautmann, 2019]

Automated Algorithm Selection on Continuous Black-Box Problems By Combining Exploratory Landscape Analysis and Machine Learning

Pascal Kerschke kerschke@uni-muenster.de
Information Systems and Statistics, University of Münster, 48149 Münster, Germany

Heike Trautmann trautmann@uni-muenster.de
Information Systems and Statistics, University of Münster, 48149 Münster, Germany

Automated Algorithm Selection - ELA + ML [Kerschke and Trautmann, 2019]

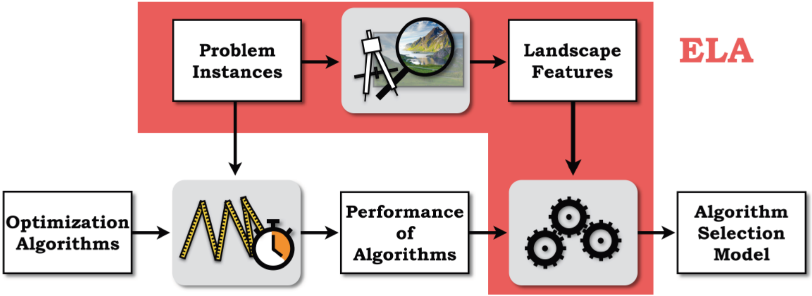


Figure 17: Schematic view of how Exploratory Landscape Analysis (ELA) can be used for improving the automated algorithm selection process. [Kerschke and Trautmann, 2019]

Automated Algorithm Selection - ELA + ML

Principle:

1. COCO BBOB benchmark utilized
2. 129 optimization algorithms performance data gathered
3. ERT computed per BBOB problem, dimension and instance
4. ELA features set from *improved latin hypercube design* $(50D)$ using *flacco* R-package
   - 102 features per problem instance (more than in [Mersmann et al., 2011])
5. 4 solvers sets (one per dimension)
   - solvers that ranked within the "Top 3" of at least one of the 24 functions
   - each set has 37 - 41 solvers
   - 12 solvers finally (optimizers that belonged to each of 4 sets)
6. Choosing best algorithm using cross-validation

Automated Algorithm Selection - ELA + ML

Algorithm selection:

- 3 algorithm selection approaches:
  - classification
  - regression
  - pairwise regression
- 4 feature selection strategies:
  - Greedy forward-backward selection (sffs)
  - Greedy backward-forward selection (sfbs)
  - (10+5)-GA
  - (10+50)-GA
- leave-one-(function)-out cross-validation
  - 96 submodels (24 functions x 4 dimensions)
  - 95 used for training + 1 for testing
- average of the resulting 96 relative ERT

Automated Algorithm Selection - ELA + ML

Results

Final models (2 of 70 = 14 algorithms x 5 feature selections):

- Model 1
  - 8 features (greedy forward-backward)
  - SVM as classification
- Model 2
  - 2 feature groups added (10+50)-GA)

Automated Algorithm Selection - ELA + ML

Results [Kerschke and Trautmann, 2019]

Dim	BBOB-Group	Relative Expected Runtime of the 12 Solvers from the Considered Algorithm Portfolio and the 2 Presented Algorithm Selection-Models													
		BSqi	BSrr	CMA-CSA	fmincon	fminunc	HCMA	HMLSL	IPOP-400D	MCS	MLSL	OQNLP	SMAC-BBOB	AS-Model #1	AS-Model #2
2	F1 - F5	**1.2**	1.3	54.8	11.0	11.8	3.7	14.6	18.4	5.8	15.5	17.0	22 014.9	16.6	20.3
	F6 - F9	18 516.7	9 708.2	7.4	18.6	19.2	5.8	1.7	5.7	11.3	24.2	**1.5**	27 518.6	3.1	3.5
	F10 - F14	7 649.2	7 481.5	8.3	**1.0**	62.7	6.3	**1.0**	10.7	322.7	**1.0**	4.9	29 353.2	4.7	4.0
	F15 - F19	7 406.6	14 710.3	14.7	7 392.0	7 367.7	25.3	8.1	15.5	7.7	7 391.7	7 351.2	29 354.8	26.2	10.1
	F20 - F24	84.8	14 768.5	7 351.9	4.1	14.5	44.9	3.9	14 679.3	11.4	**2.1**	2.7	22 014.6	42.5	3.0
	all	6 240.7	9 318.4	1 549.1	1 546.5	1 556.7	17.7	**6.0**	3 068.4	74.3	1 547.9	1 536.9	25 990.1	19.3	8.4
3	F1 - F5	**1.3**	1.3	7 367.9	85.2	132.1	356.1	6.8	14 686.6	45.9	55.9	7 347.6	22 015.1	58.4	94.9
	F6 - F9	331.2	9 527.4	4.7	38.5	9 173.7	4.5	**1.9**	6.5	31.4	9 173.4	2.5	36 690.3	3.3	39.9
	F10 - F14	29 356.3	14 712.1	8.9	**1.0**	4.1	5.0	1.0	12.3	8 132.7	1.0	9.3	29 353.4	4.8	3.6
	F15 - F19	14 698.2	22 026.2	**1.6**	14 701.2	14 699.5	2.6	11.4	7 339.4	7 346.9	14 700.0	14 686.2	36 690.3	2.8	7.1
	F20 - F24	14 741.8	14 758.7	7 389.4	7 339.6	14 677.4	66.8	2.3	22 015.1	7 342.4	7 339.8	**1.9**	22 014.8	67.0	3.4
	all	12 304.7	12 316.7	3 077.4	4 616.2	7 677.5	90.4	**4.8**	9 178.9	4 769.4	6 132.4	4 593.1	29 047.1	28.3	29.4
5	F1 - F5	**1.4**	1.4	7 533.6	14 678.4	14 679.2	12.0	17.5	14 688.7	14 678.1	14 678.5	14 678.0	22 015.1	22.7	22.9
	F6 - F9	27 597.4	36 690.3	5.6	9 173.5	9 173.8	3.9	**2.4**	4.9	28.8	9 173.4	9 173.5	36 690.3	4.8	4.8
	F10 - F14	22 032.8	29 360.3	8.9	1.0	11.9	4.2	**1.0**	13.6	22 019.2	**1.0**	10.7	36 690.3	5.2	5.2
	F15 - F19	36 690.3	36 690.3	**3.1**	36 690.3	36 690.3	4.3	7 346.1	29 352.5	36 690.3	36 690.3	29 352.5	36 690.3	4.4	4.4
	F20 - F24	22 053.6	22 050.8	7 400.0	14 678.9	22 014.9	7.7	7 339.8	22 017.4	14 681.0	22 015.0	14 676.8	22 014.9	7.8	7.8
	all	21 428.3	24 469.8	3 114.6	15 289.0	16 819.9	**6.5**	3 063.8	13 765.8	18 352.4	16 817.4	13 761.8	30 575.6	9.1	9.2
10	F1 - F5	**1.6**	1.6	14 691.0	14 679.9	14 682.7	2.7	7 365.5	14 698.8	14 680.0	14 679.9	14 678.3	22 015.7	16.3	16.3
	F6 - F9	36 690.3	27 563.9	4.3	9 173.4	9 173.8	2.2	4.1	9 181.9	9 188.1	9 173.4	9 173.9	36 690.3	2.7	2.7
	F10 - F14	29 359.3	29 359.8	8.4	1.1	15.4	2.8	**1.1**	7 352.5	22 018.7	**1.1**	12.0	36 690.3	3.7	3.7
	F15 - F19	36 690.3	36 690.3	**1.7**	36 690.3	36 690.3	2.0	22 028.5	29 352.5	36 690.3	36 690.3	36 690.3	36 690.3	2.1	2.1
	F20 - F24	36 690.3	29 367.0	14 685.9	22 015.2	22 015.0	**23.6**	14 677.1	29 352.8	22 018.9	22 014.6	22 014.9	36 690.3	23.7	23.7
	all	27 519.5	24 472.9	6 123.0	16 817.8	16 821.3	**6.9**	9 182.4	18 354.6	21 408.0	16 817.6	16 819.7	33 633.1	10.0	10.0
all	F1 - F5	**1.4**	1.4	7 411.8	7 363.6	7 376.5	93.6	1 851.1	11 023.1	7 352.4	7 357.4	9 180.2	22 015.2	28.5	38.6
	F6 - F9	20 783.9	20 872.4	5.5	4 601.0	6 885.1	4.1	**2.5**	2 299.8	2 314.9	6 886.1	4 587.9	34 397.4	3.5	12.7
	F10 - F14	22 099.4	20 228.4	8.7	1.0	23.5	4.6	**1.0**	1 847.3	13 123.3	**1.0**	9.3	33 021.8	4.6	4.1
	F15 - F19	23 871.3	27 529.3	**5.2**	23 868.5	23 861.9	8.6	7 348.5	16 515.0	20 183.8	23 868.1	22 020.0	34 856.4	8.9	5.9
	F20 - F24	18 392.6	20 236.3	9 206.8	11 009.4	14 680.5	**35.8**	5 505.8	22 016.2	11 013.4	12 842.9	9 174.1	25 683.7	35.3	9.5
	all	16 873.3	17 644.5	3 466.0	9 567.4	10 718.9	**30.4**	3 064.3	11 091.9	11 151.0	10 328.8	9 177.9	29 811.5	*16.7*	*14.2*

Automated Algorithm Selection - ELA + ML

Results

True Best Solver		Predicted Solver (Model 1 / Model 2)			
Solver	#	fmincon	HCMA	HMLSL	MLSL
BSqi	6	3 / 2	3 / 2	0 / 2	0 / 0
BSrr	6	1 / 2	5 / 4	0 / 0	0 / 0
CMA-CSA	7	0 / 1	7 / 3	0 / 3	0 / 0
fmincon	12	0 / 4	8 / 4	0 / 1	4 / 3
fminunc	6	1 / 2	4 / 3	0 / 0	1 / 1
HCMA	14	0 / 3	14 / 11	0 / 0	0 / 0
HMLSL	11	3 / 3	7 / 4	0 / 0	1 / 4
IPOP400D	7	0 / 0	7 / 3	0 / 3	0 / 1
MCS	4	2 / 1	2 / 0	0 / 3	0 / 0
MLSL	12	4 / 2	8 / 6	0 / 2	0 / 2
OQNLP	6	0 / 1	6 / 2	0 / 2	0 / 1
SMAC-BBOB	5	0 / 0	5 / 2	0 / 1	0 / 2
Σ	96	14 / 21	76 / 44	0 / 17	6 / 14

Figure 18: Comparison of the predicted solvers. Each row shows how often the respective best solver was predicted as fmincon, HCMA, HMLSL or MLSL by the selectors (Model 1 / Model 2). [Kerschke and Trautmann, 2019]

Conclusions

- ELA is effective in function class prediction
- ELA is useful in Algorithm Selection
- Generalizability of ELA is unconvincing

Bibliography

Bischl, B., Kerschke, P., Kotthoff, L., Lindauer, M., Malitsky, Y., Fréchette, A., Hoos, H., Hutter, F., Leyton-Brown, K., Tierney, K., et al. (2016).
Aslib: A benchmark library for algorithm selection.
Artificial Intelligence, 237:41–58.

Hansen, N., Brockhoff, D., Mersmann, O., Tusar, T., Tusar, D., ElHara, O. A., Sampaio, P. R., Atamna, A., Varelas, K., Batu, U., Nguyen, D. M., Matzner, F., and Auger, A. (2019).
COmparing Continuous Optimizers: numbbo/COCO on Github.

Kerschke, P., Hoos, H. H., Neumann, F., and Trautmann, H. (2019).
Automated algorithm selection: Survey and perspectives.
Evolutionary computation, 27(1):3–45.

Kerschke, P. and Trautmann, H. (2019).
Automated algorithm selection on continuous black-box problems by combining exploratory landscape analysis and machine learning.
Evolutionary computation, 27(1):99–127.

Mersmann, O., Bischl, B., Trautmann, H., Preuss, M., Weihs, C., and Rudolph, G. (2011).
Exploratory landscape analysis.
In *Proceedings of the 13th annual conference on Genetic and evolutionary computation*, pages 829–836.

Mersmann, O., Preuss, M., and Trautmann, H. (2010).
Benchmarking evolutionary algorithms: Towards exploratory landscape analysis.
In *International Conference on Parallel Problem Solving from Nature*, pages 73–82. Springer.

Rice, J. R. (1976).
The algorithm selection problem.
In *Advances in computers*, volume 15, pages 65–118. Elsevier.

Tanabe, R. (2021).
Towards exploratory landscape analysis for large-scale optimization: a dimensionality reduction framework.
arXiv preprint arXiv:2104.10301.

Wolpert, D. H., Macready, W. G., et al. (1995).
No free lunch theorems for search.
Technical report, Technical Report SFI-TR-95-02-010, Santa Fe Institute.