Prof. Dr. Pascal Kerschke
TU Dresden, “Friedrich List” Faculty of Transport and Traffic Sciences, Big Data Analytics in Transportation, and Center for Scalable Data Analytics and Artificial Intelligence (ScaDS.AI) Dresden/Leipzig

Automated Algorithm Selection

Summer School on Automatic Algorithm Design 2023 // Thursday, June 15, 2023, in Lille, France

Thanks to the Contributions of Many Smart Minds ☺







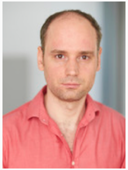

















Academic Background

- TU Dortmund University, Dortmund, Germany
    - 2007 – 2010: B.Sc. *Data Analysis and Management*
    - 2010 – 2013: M.Sc. *Data Science*
- University of Münster, Münster, Germany
    - 2013 – 2017: Ph.D. in *Information Systems* (Supervisor: Heike Trautmann)
    - 2017 – 2021: PostDoc at *Data Science: Statistics & Optimization* (head: Heike Trautmann)
- TU Dresden, Dresden, Germany
    - since 2021: Professor for *Big Data Analytics in Transportation*

Where is Dresden?

Travel Times (Train):

- ca. 2h to Berlin
- ca. 2.5h to Prague
- ca. 4h to Frankfurt
- ca. 4h to Hamburg
- ca. 5h to Munich
- ca. 6h to Dortmund
- ca. 9h to Bruxelles
- ca. 10h to Paris
- ca. 11h to Lille

My Wonderful Team ☺

Felix Rauschert

Lennart Schäpermeier

Konstantin Dietrich

Sandy Wolf

Markus Leyser

Jonathan Heins

Felix Hoch

Our Research

Research Interests:

- Automated Algorithm Selection
- Benchmarking
- Data Science
- Exploratory Landscape Analysis
- Optimization:
    - Single-Objective Continuous Optimization
    - Multi-Objective Continuous Optimization
    - Routing Problems (TSP, VRP)
- Machine Learning (AutoML, IML/XAI; Unsupervised, Supervised, and Reinforcement Learning)
- Visualization of Optimization Problems

Research Networks:

- Benchmarking Network
- COSEAL (Configuration and Selection of Algorithms)
- CLAIRE (Confideration of Laboratories for Artificial Intelligence Research in Europe)
- Center for Scalable Data Analytics and Artificial Intelligence (ScaDS.AI) Dresden/Leipzig
- Boysen-TU Dresden-Research Training Group

Prof. Dr. Pascal Kerschke
TU Dresden, “Friedrich List” Faculty of Transport and Traffic Sciences, Big Data Analytics in Transportation, and Center for Scalable Data Analytics and Artificial Intelligence (ScaDS.AI) Dresden/Leipzig

Automated Algorithm Selection

Summer School on Automatic Algorithm Design 2023 // Thursday, June 15, 2023, in Lille, France

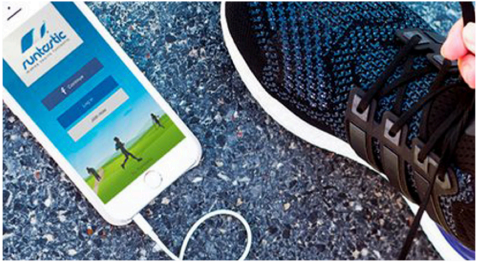
Agenda

- Introduction
  - Optimization
  - Traveling Salesperson Problem
  - Automated Algorithm Selection
- Building Blocks of Automated Algorithm Selection
- Status Quo of Automated Algorithm Selection
  - Traveling Salesperson Problem
  - Single-Objective Continuous Optimization
- Benefits and Perspectives

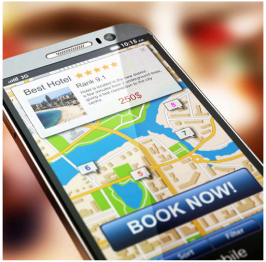
Introduction

www.dailymail.co.uk


www.autobild.de

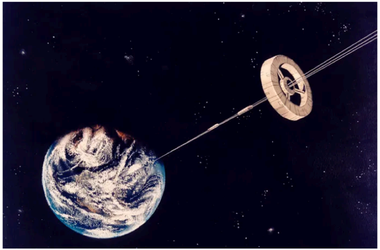


digitalhealthage.com



www.aktiv-online.de

internetofbusiness.com



www.spacelegalissues.com

www.newsmax.com

www.autonomousvehicleinternational.com

Introduction

Optimization

Optimization

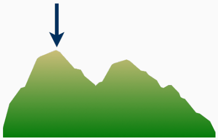
- ideally, such problems can be expressed as a mathematical function $f : \mathcal{X} \to \mathbb{R}$
- full knowledge enables analytical investigations of the underlying model

input → $\boxed{\textbf{model}}$ → **output**

- in such a case, one is interested in x_{opt} such that (w.l.o.g.)

$$x_{opt} = \arg\max_{x \in \mathcal{X}} f(x)$$

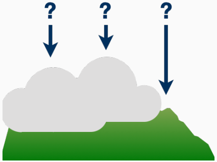
- optimization is quite simple



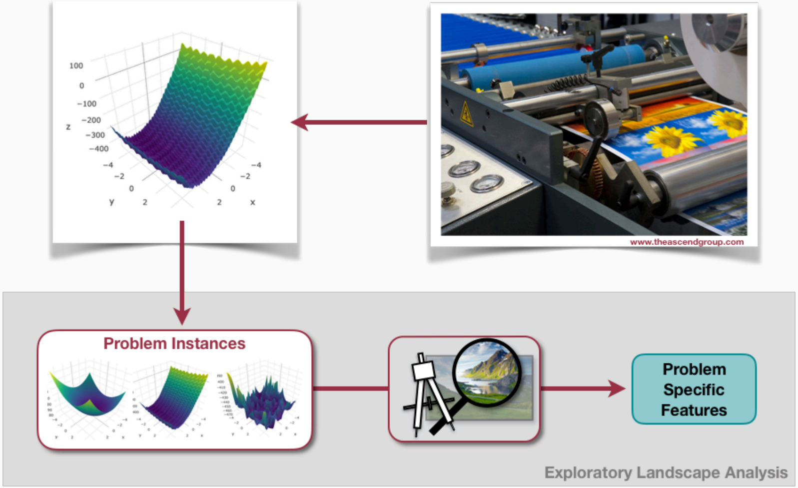
Black-Box Optimization

- unfortunately, in most real-world cases, such mathematical function are unknown
- underlying models become *black-box models*

- x_{opt} has to be determined by sophisticated algorithms that do not need knowledge about the model
- optimization becomes much more challenging



- extract information from (black-box) optimization problems in an automated fashion



Feature-Based Problem Characterization

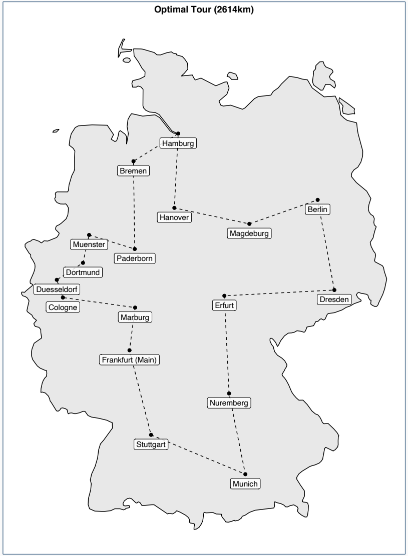
- features are simple (numerical) statistics that “describe” a problem’s structure
- computed based on
  1. continuous problems:
     a “small” sample of n evaluated points $x_{i1}, \dots, x_{id}, i = 1, \dots, n$
     (including their corresponding fitness value y_i)
  2. graph-based problems (e.g., TSP):
     characteristics of a graph (amount of nodes, edge lengths, etc.)
- features help to improve understanding of
  - problems (match features to high-level properties)
  - algorithm-to-problem dependencies
- nowadays, “ELA” is frequently used as synonym for feature-based characterization of continuous single-objective (black-box) optimization problems

Introduction

Traveling Salesperson Problem

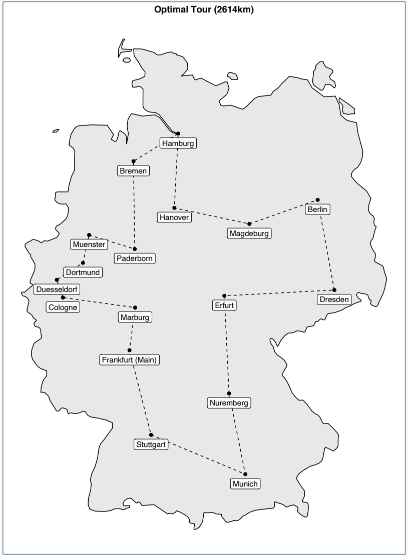
Traveling Salesperson Problem

- classical NP-hard optimization problem
- studied for decades
- important for various applications:
  - transportation logistics
  - circuit board fabrication
  - etc.
- given:
  - a set of cities
  - edges connecting the cities
- goal:
  - find the shortest round trip such that every city is visited exactly once



Optimizing Traveling Salesperson Problems

- optimization algorithms distinguished into
  - exact algorithms
  - inexact heuristics
- exact algorithms:
  - guarantee optimality of found solutions
  - oftentimes expensive / slow
  - state of the art: Concorde [3]
- inexact heuristics
  - usually much faster
  - solution quality (i.e., length of found tours) is very often (at least close to) optimal
  - state of the art are restart variants [11] of:
    - Lin-Kernighan Heuristic (LKH) [18]
    - Genetic Algorithm using Edge Assembly Crossover (EAX) [37]



Introduction

Automated Algorithm Selection

Motivation of Automated Algorithm Selection

- *No Free Lunch Theorem* [51]:
  "better" optimization algorithm depends on problem structure
- similar to choice of preferred vehicle
- don't just rely on "gut-feeling"
- use additional information / data and automated techniques
- *Automated Algorithm Selection*:
  - introduced by John Rice in 1976 [43]
  - combines problem knowledge and (performance) data
  - relies on techniques from (automated) machine learning


https://taz.de/


https://www.wn.de/


https://www.mein-schwarzwald.de/


https://www.bvg.de/


https://finanzwelt.de/

Automated Algorithm Selection – Survey Articles

[48] Kate A. Smith-Miles (2009).

Cross-Disciplinary Perspectives on Meta-Learning for Algorithm Selection. ACM Computing Surveys, 41:1-25.

- meta-learning perspective
- covers developments from perspective of AI and operations research communities

[26] Lars Kotthoff (2014).

Algorithm Selection for Combinatorial Search Problems: A Survey. AI Magazine, 35(3):48-60.

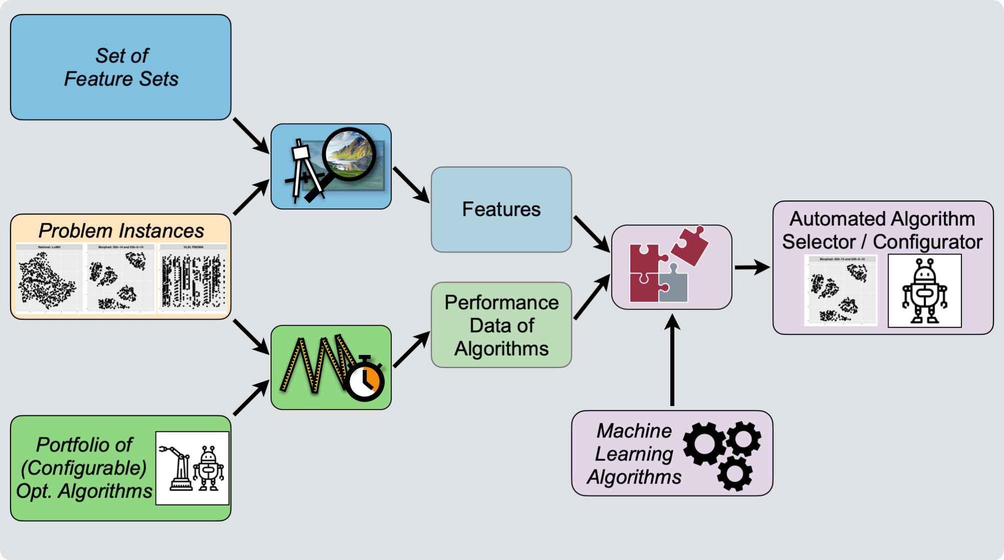
- focus on discrete problems
- explains several aspects of algorithm selection (e.g., online vs. offline selection, static vs. dynamic portfolios, consideration of costs, different prediction types)

[25] Pascal Kerschke, Holger Hoos,
Frank Neumann & Heike Trautmann (2019).
Automated Algorithm Selection:
Survey and Perspectives.
Evolutionary Computation, 27(1):3-45, MIT Press.

- most recent survey
- focus on feature-based per-instance algorithm selection (PIAS)
- connection to related problems (algorithm configuration, schedules, etc.)
- dedicated section on perspectives for future research



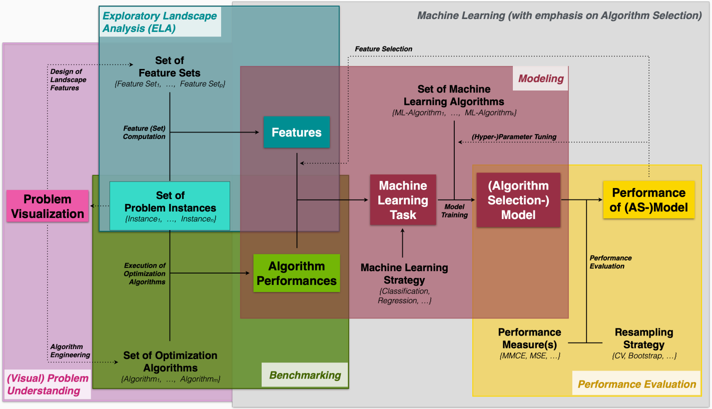

General Idea of Automated Algorithm Selection



Building Blocks of Automated Algorithm Selection

Building Blocks of Automated Algorithm Selection

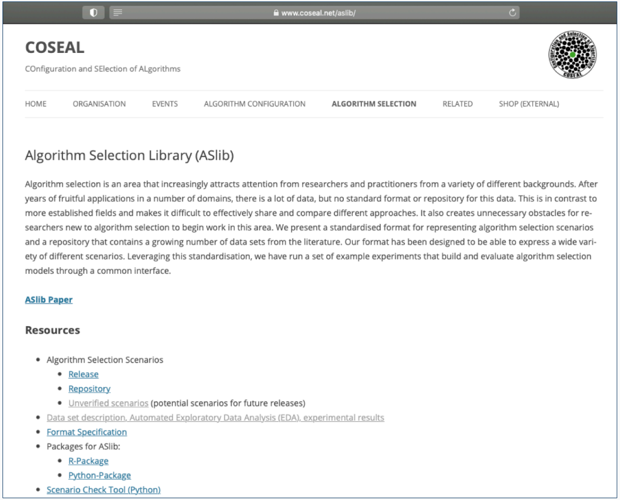
- a good AS model relies on various components [25]:
  - representative problem instances
  - good understanding of problem characteristics
  - cheap and informative features
  - portfolio of competitive and complementary algorithms
  - powerful machine learning algorithms



  - suitable performance measure (w.r.t. optimization algorithm and AS model)
- frequently confronted with various forms of big data

Algorithm Selection Library

- data of numerous AS scenarios can be found in the Algorithm Selection Library (ASLib [6]):
  - initiated by members of the COSEAL network (`coseal.net`)
  - unified format specifications
  - currently contains ≈30 scenarios
  - offers interfaces to python and R
  - overview per scenario
  - validation tool



www.coseal.net/aslib/

COSEAL

COnfiguration and SElection of ALgorithms

HOME ORGANISATION EVENTS ALGORITHM CONFIGURATION **ALGORITHM SELECTION** RELATED SHOP (EXTERNAL)

Algorithm Selection Library (ASlib)

Algorithm selection is an area that increasingly attracts attention from researchers and practitioners from a variety of different backgrounds. After years of fruitful applications in a number of domains, there is a lot of data, but no standard format or repository for this data. This is in contrast to more established fields and makes it difficult to effectively share and compare different approaches. It also creates unnecessary obstacles for researchers new to algorithm selection to begin work in this area. We present a standardised format for representing algorithm selection scenarios and a repository that contains a growing number of data sets from the literature. Our format has been designed to be able to express a wide variety of different scenarios. Leveraging this standardisation, we have run a set of example experiments that build and evaluate algorithm selection models through a common interface.

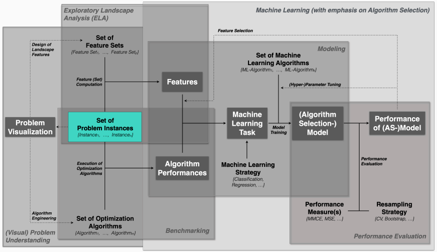
ASlib Paper

Resources

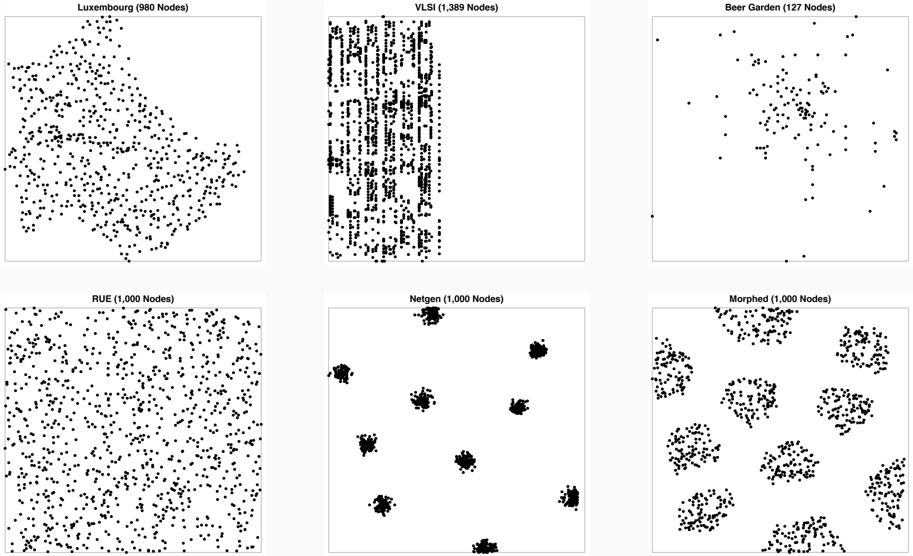
- Algorithm Selection Scenarios
  - Release
  - Repository
  - Unverified scenarios (potential scenarios for future releases)
- Data set description, Automated Exploratory Data Analysis (EDA), experimental results
- Format Specification
- Packages for ASlib:
  - R-Package
  - Python-Package
- Scenario Check Tool (Python)

Building Blocks of AAS – Instances

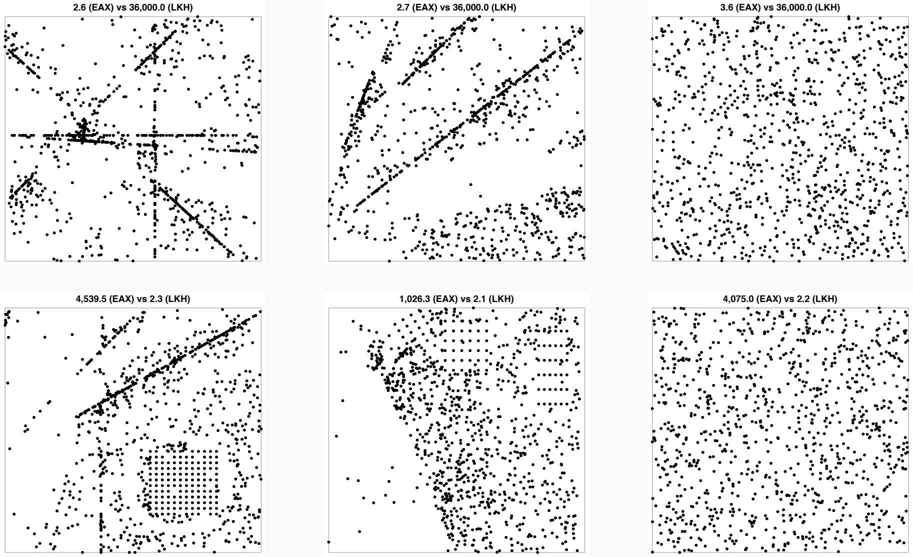
- problem instances of an optimization benchmark
  - possess various “landscape” characteristics, which could potentially be obstacles for certain algorithms
  - are capable of revealing strengths and weaknesses of optimization algorithms
  - provide a playground for simple as well as complex algorithms
  - are representative for real-world problems
- their representation affects the choice of optimization algorithms, features, etc.



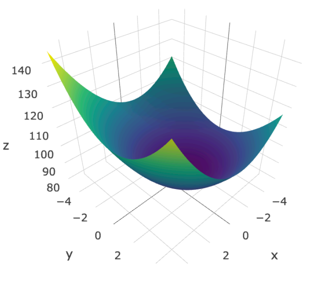
Building Blocks of AAS – Instances: TSP I

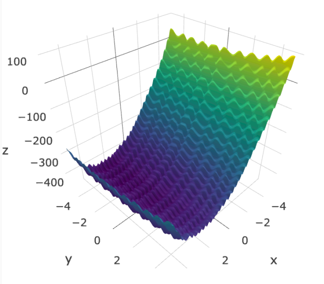


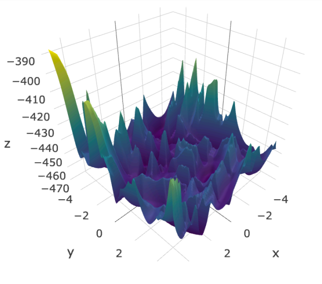
Building Blocks of AAS – Instances: TSP II

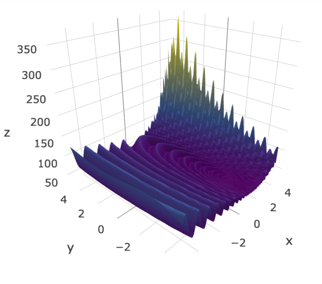


Building Blocks of AAS – Instances: Continuous Optimization



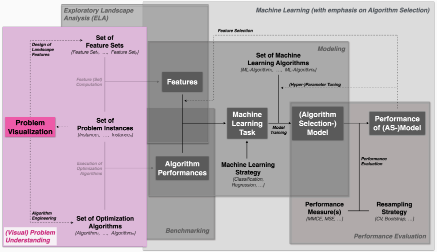




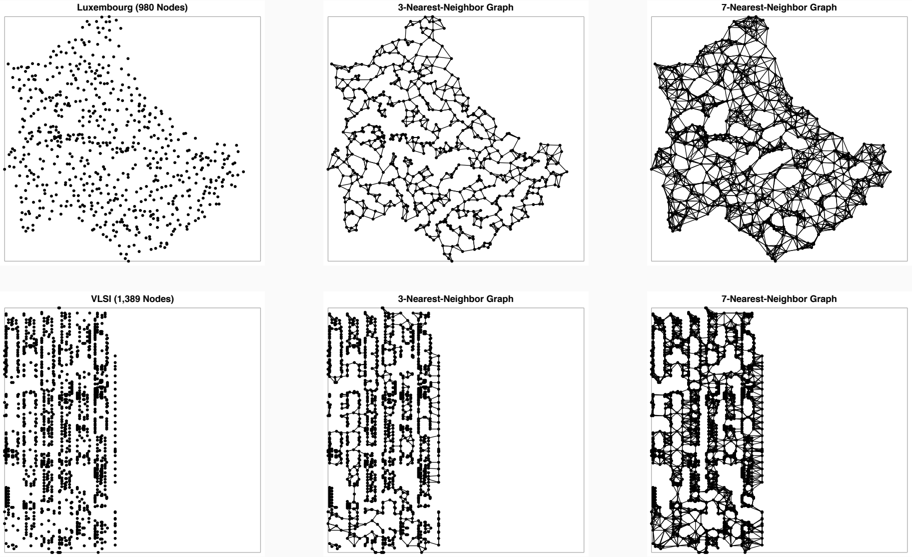


Building Blocks of AAS – Problem Visualization

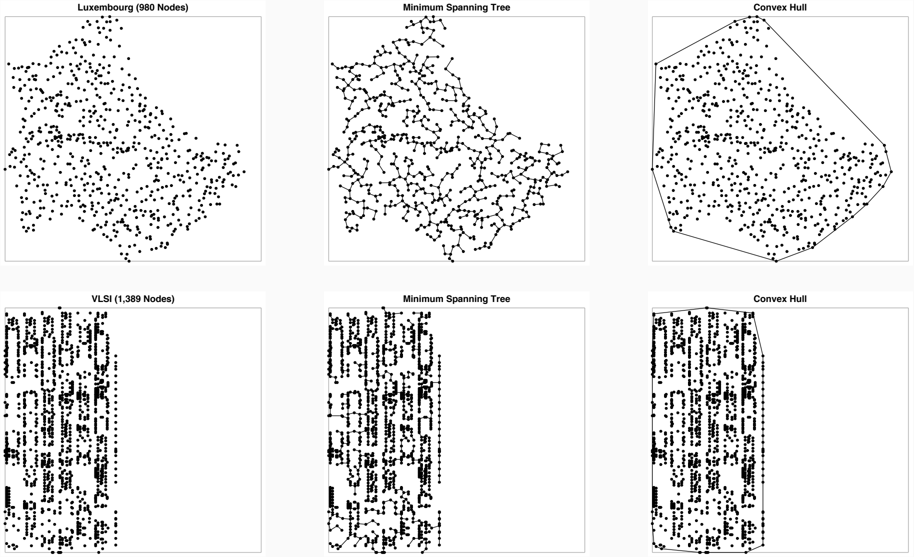
- purpose of problem visualization:
  - get a “feeling” for / better understanding of the domain’s (benchmark) problems
  - look for problem properties, which likely also exist in higher-dimensional (i.e., non-visualizable) versions
  - identify potential pitfalls (“traps”) for optimization algorithms $\rightsquigarrow$ use those findings for improved algorithm engineering



Building Blocks of AAS – Visualization: TSP I

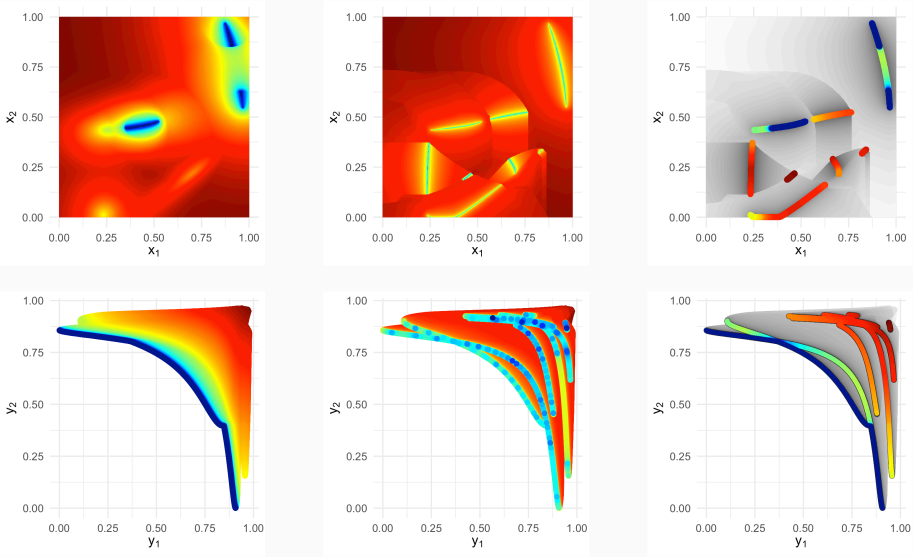


Building Blocks of AAS – Visualization: TSP II

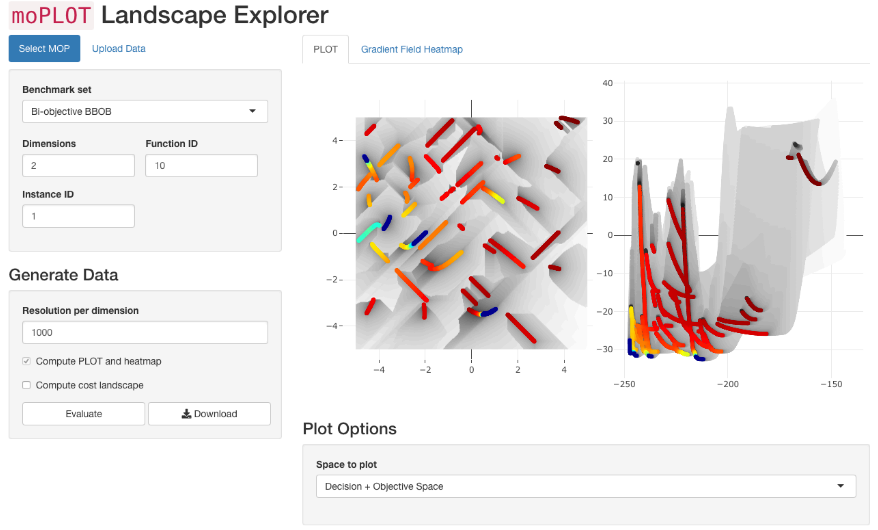


(a) **Ackley**, $pstep = 2\beta$ (b) **Rastrigin**, $pstep = 2\beta$ (c) **Birastrigin**, $pstep = 2\beta$

$\rightsquigarrow$ *Local Optima Networks for Continuous Fitness Landscapes* [1]


1.00
0.75
0.50
0.25
0.00
0.00 0.25 0.50 0.75 1.00
x_1
x_2
y_1
y_2

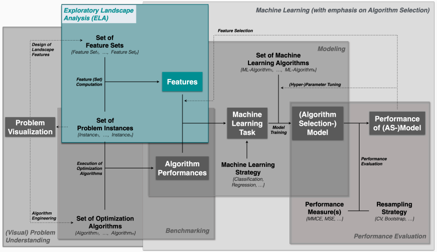
Building Blocks of AAS – Visualization: Multi-Objective Optimization II



⤳ make use of interactive tools such as **moPLOT** [46]

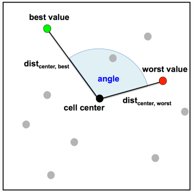
Building Blocks of AAS – Features

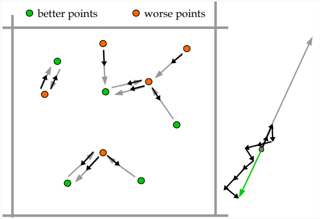
- features (i.e., numeric low-level statistics) capture a problem's essential information
- as informative and cheap as possible
- describe problem properties and/or can be used to discriminate between algorithms
- design of features (but not their practical application) requires domain-knowledge
- for various optimization domains, there exist either precomputed features or tools to compute them
- enables interpretation of algorithm selector's decision making procedure

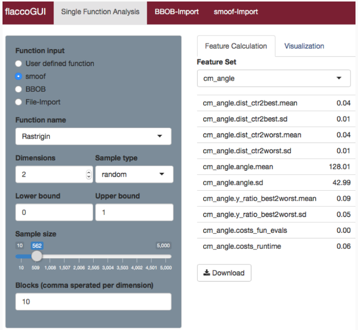


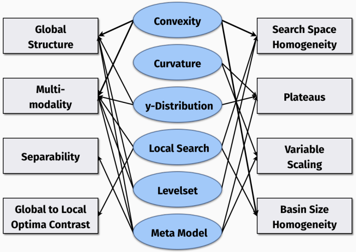
Building Blocks of AAS – Features: Continuous Optimization I

- mainly based on a small sample of (evaluated) points from the problem's landscape
- numerous research groups from all over the world developed hundreds of features
- majority of features available in R-package `flacco` [22], web-application `flaccoGUI` [15], and python-package `pflacco` [41]

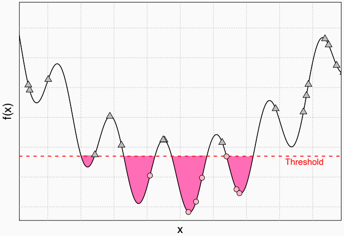




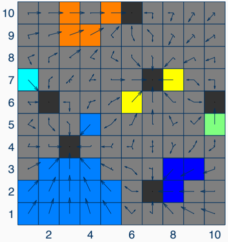




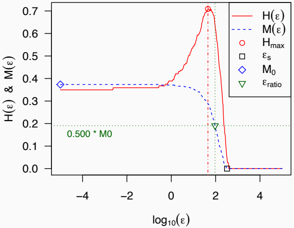
Building Blocks of AAS – Features: Continuous Optimization II



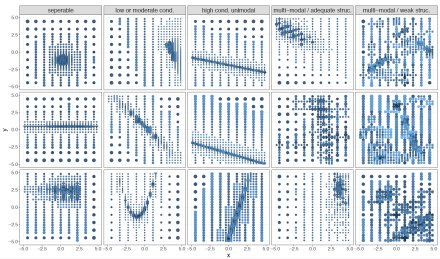
[32]



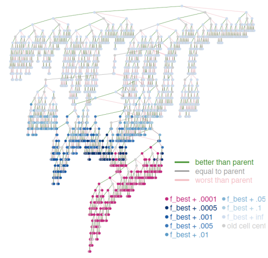
[23]



[36]

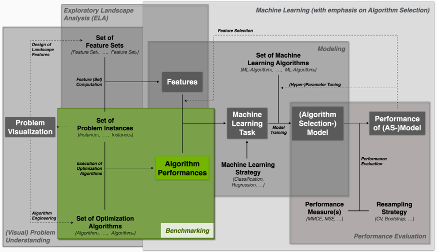


[9]

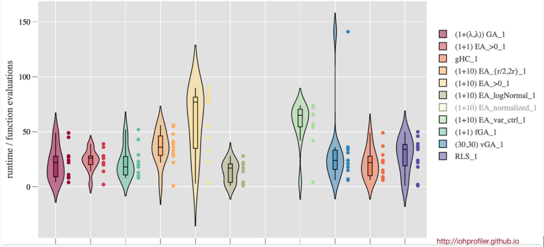


Building Blocks of AAS – Benchmarking

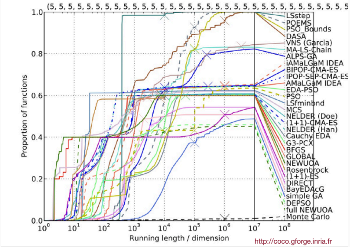
- measures performance of an algorithm (usually runtime or solution quality)
- which algorithms are promising, useless, etc.?
- which problems are easy, challenging, etc.?
- in some domains, large data bases exist [13]
- survey paper on benchmarking optimization algorithms [4]
- investigate how much each algorithm contributes to the portfolio



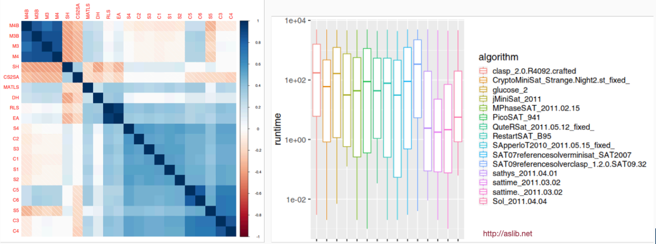
Building Blocks of AAS – Benchmarking: Tools and Approaches



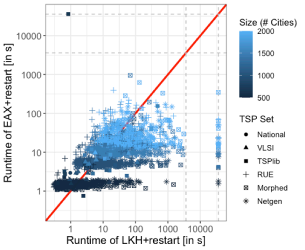
[10]



[13]



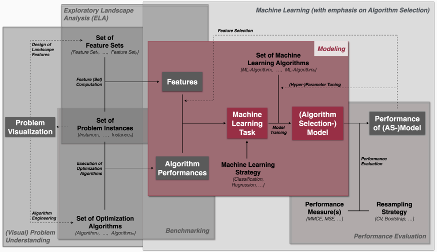
[6]



[24]

Building Blocks of AAS – Modeling

- map problem features to algorithm performances
- different approaches:
  - classification: ‘correct’ class is the best algorithm (per instance)
  - regression: model performance per algorithm, and select (per instance) the algorithm with the best predicted performance
  - pairwise regression
  - clustering
  - classifier chains
- quality of the selector highly depends on considered (subset of) features, choice of ML algorithm, its (hyperparameter) configuration



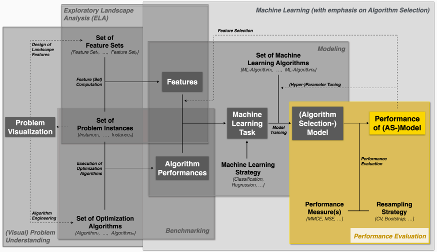
Building Blocks of AAS – Modeling: ML Frameworks

- there exist numerous tools for performing such ML tasks, such as
  - R: **`mlr3`** [28] (https://mlr3.mlr-org.com)
  - python: **`scikit-learn`** [38] (https://scikit-learn.org/stable/)
- algorithm selector attempts to close the gap between a portfolio's SBS (0%) and VBS (100%):
  - *single best solver (SBS)*: best stand-alone algorithm from the portfolio (based on each algorithm's average performance across the whole data set)
  - *virtual best solver (VBS)*: always picking the best algorithm from the portfolio "for free" (oracle-like prediction per instance)
- the SBS and VBS provide the lower and upper limits of the AS performance
- VBS-like performance is unlikely (feature computation is not for free)

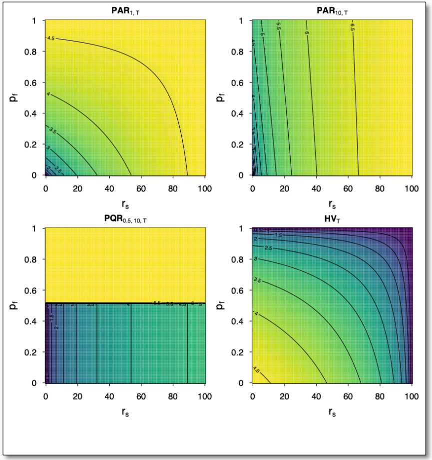
Problem	Algorithm			VBS
	A	B	C	
P1	1	5	2	1
P2	1500	1	1	1
P3	50	150	1	1
...	...	...	...	...
P50	8	1	10	1
∅	34	10	3	1
			SBS	

Building Blocks of AAS – Performance

- avoid overfitting by using a resampling strategy (crossvalidation, bootstrap sampling, holdout, etc.)
- use an appropriate performance measure
  - accuracy (often the default in classification) $\rightsquigarrow$ treats all errors equally
  - oftentimes, the misclassification penalty is a better choice as it can help the selector to focus on the relevant cases (where misclassifications are actually costly)
- pass the selector's performance to the previous (modeling) stage to find a better selector



- one needs one scalar value for each pair of algorithm and problem instance
- need to handle information of repeated runs (to assess algorithm stochasticity)
- need to treat performance of failed runs (e.g., penalized by 10-fold of maximum budget)
- aggregate performance across all runs (per pair of algorithm and problem instance)
- typical measures (depending on the domain):
  - penalized average runtime (e.g., PAR10 or PAR2)
  - penalized quantile runtime (e.g., PQR10)
  - expected runtime



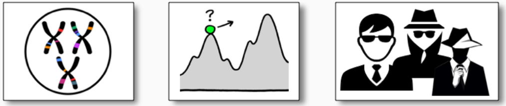
Status Quo of Automated Algorithm Selection

Status Quo of Automated Algorithm Selection

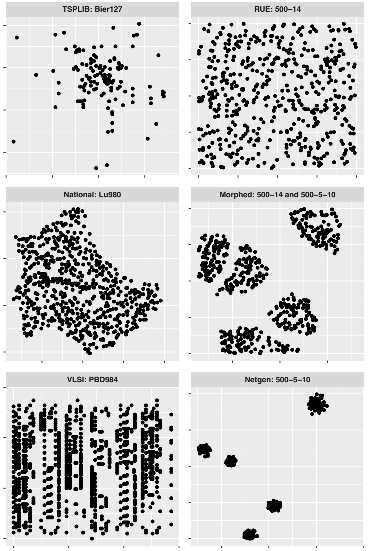
Traveling Salesperson Problem

Automated Algorithm Selection in TSP

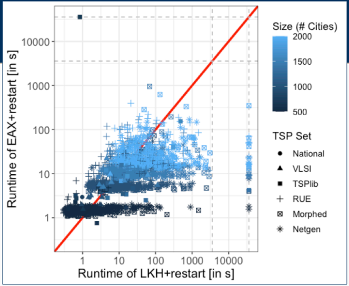
- most comprehensive AS study for TSP [24]
- six TSP sets with 1 845 instances in total:
  - ‘real-world’ TSP sets: TSPLib, National, VLSI
  - artificial TSP sets: RUE, Netgen, Morphed
- five state-of-the-art solvers:
  - EAX [37]
  - LKH [18]
  - restart variants of EAX and LKH [11]
  - MAOS [52]

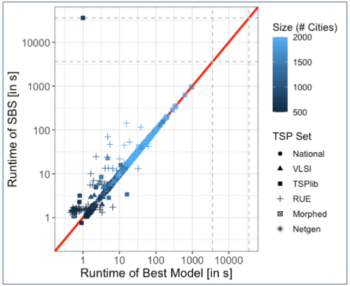


- feature sets (with 400+ features):
  - TSPmeta [35]
  - UBC [20]
  - Pihera [39]



- measured performance via PQR10 scores[1]
- baselines for performance assessment:
  - single-best solver (SBS): 36.30s (here: EAX+restart)
  - virtual best solver (VBS): 10.73s
  - 2nd-best solver (LKH+restart): 565.85s
- trained 14 candidate AS models per feature set
  - decision trees, random forests, SVMs, MARS models
  - classification, regression, paired regression
- considered 4 feature selection strategies per selector
- best found selector:
  - classification-based SVM
  - uses 16 of the 400+ features
  - PQR10-score: 16.75s (closed 75% of SBS-VBS gap)

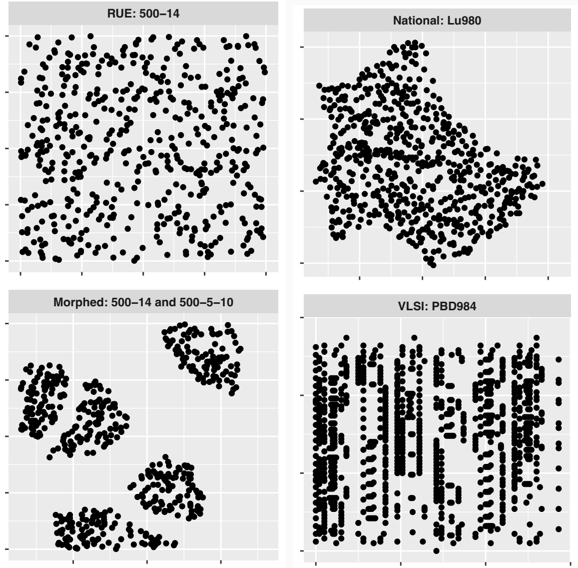




1. $\mathrm{PQR}_{p,f,T}(A,I) = \begin{cases} f \cdot T, & \text{if } \sum_{i=1}^{m} \mathbb{1}\{r_i^{A,I} < T\} < \lfloor m \cdot p + 1 \rfloor \\ q_p(r_1^{A,I}, \dots, r_m^{A,I}), & \text{otherwise} \end{cases}$

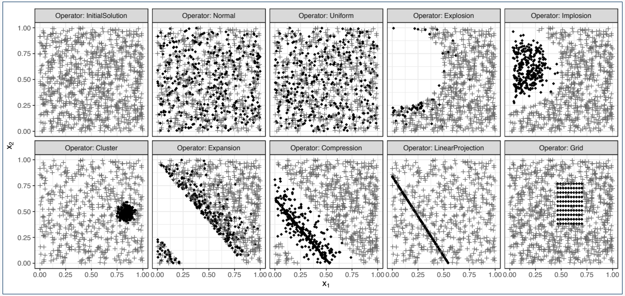
Gap Between Artificial and Real-World Problems

- set of representative and discriminative TSP instances is essential
- limited amount of real-world instances
- instance generators create very artificial problems

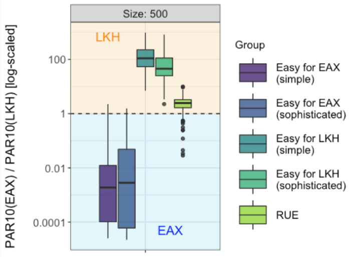


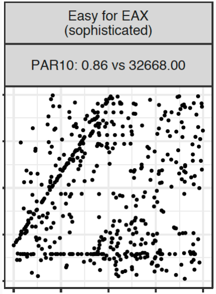
Bridging the Gap

- designed "more realistic" mutation operators [7] capable of creating more structural instances



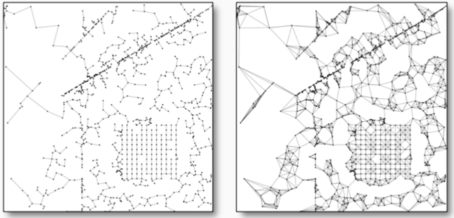
- evolved TSP instances that were tailored to be easy for EAX and challenging for LKH (and vice versa)

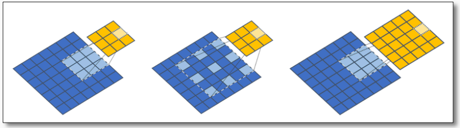




Deep Learning as Feature-Free Alternative

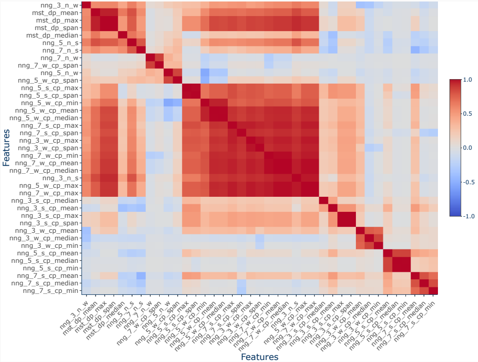
- features need to be cheaply computable, informative and discriminative
- investigated applicability of deep learning as “feature-free” alternative [47]
- images of TSP instances as input
    - scatterplot of cities
    - minimum spanning tree
    - k-nearest neighbor graph (with $k = 5$)
- different convolutional neural networks
- closed SBS-VBS-gap:
    - classical AS approach: 17.9%
    - CNN based on cities: 17.8%
    - CNN based on cities + MST: 21.5%



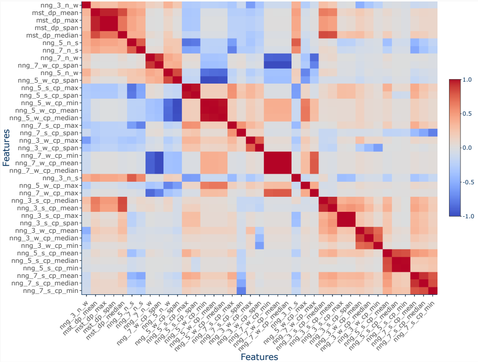


Robustness of TSP Features

- many TSP features are affected by instance size or search space dimensionality
- recent studies examined the potential of normalization [16, 17]



without normalization



with normalization

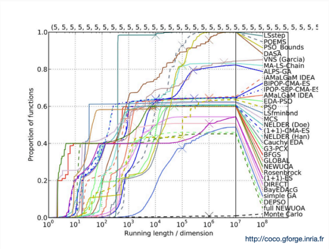
- dashboard at: https://tsp-features.shinyapps.io/normalization-eda/

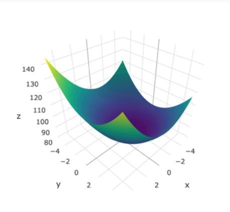
Status Quo of Automated Algorithm Selection

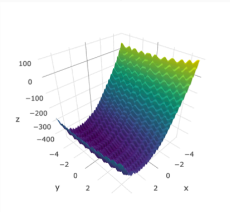
Continuous Black-Box Optimization

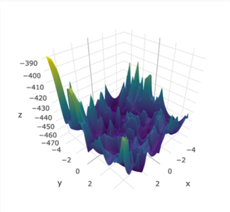
Automated Algorithm Selection in Continuous Black-Box Optimization

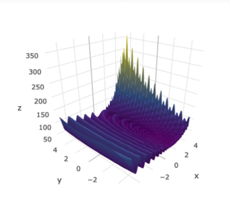
- conducted the up to now most comprehensive AS study for single-objective continuous optimization [21]
- experiments based on a total of 480 instances from the Black-Box Optimization Benchmark (BBOB) [12]
- considered performance data of all 129 optimization algorithms that were available in COCO [14]




Sphere

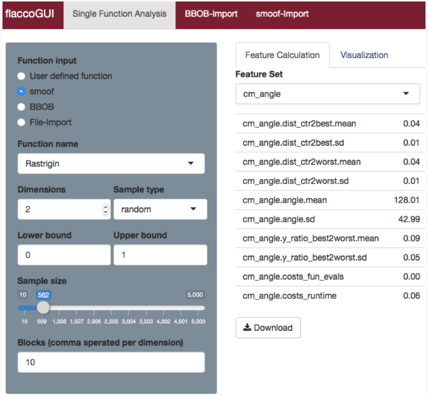

Rastrigin


Gallagher's Gaussian 101-me Peaks


Schaffers F7

Exploratory Landscape Analysis

- *Exploratory Landscape Analysis (ELA)*
  - research area focussing on characterization of continuous optimization problems
  - seminal work [34] has been acknowledged with the *ACM SigEVO Impact Award 2021*
- characterized problems by means of 100+ features using the R-package **`flacco`** [22]
- **`flacco`** is also available for non-R-users:
  - python-package **`pflacco`** [40]
  - platform-independent web application [15] (https://flacco.shinyapps.io/flacco/)



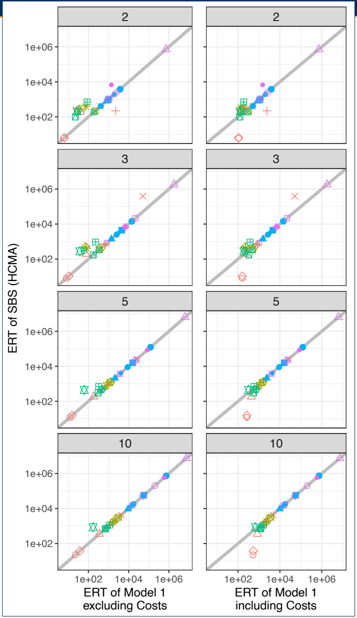
Automated Algorithm Selection in Continuous Black-Box Optimization

- reduced portfolio from 129 to 12 optimizers:



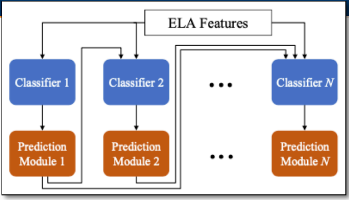

2x deterministic 5x multi-level 4x CMA-ES 1x commercial

- measured performance as PAR10 score of relative expected runtime (relERT)
  $\rightsquigarrow$ virtual best solver (VBS) has to be 1
- hybrid algorithm HCMA [31] is single-best solver with a mean PAR10 score of 30.4
- best found AS model: SVM (using 9 features) with a mean PAR10 score of 14.2



Towards Automated Algorithm Configuration in Continuous Optimization

- feature-based approaches are so far mainly used in AS (but not in AC)
- modularized CMA-ES [49]
  - 9 binary and 2 trinary modules
  - in total, $2^9 \times 3^2 = 4\,608$ configurations possible
  - SBS among all 4 608 configurations was on average 28 times slower than the VBS
- classifier chains in combination with ELA features for landscape-aware AC [42]
  - order of the 11 modules matters
  - 11! (≈40 mio.) classifier chains were investigated
- best classifier chain is competitive
  - relative ERT of 16.02 (40% speed-up over SBS)
  - sometimes even outperformed HCMA



		Baseline			CC		COCO	
D	Function	SBS	CMA	IPOP	no	with	HCMA	BRENT
2	01-05	110.12	401.30	106.11	42.09	48.25	0.55	0.27
	06-09	20.65	17.30	3.75	6.56	8.53	2.80	7920.81
	10-14	2.96	3.69	5.09	1.77	3.27	1.53	3065.59
	15-19	6.81	417.56	5.95	21.16	22.11	5.16	2478.98
	20-24	116.69	372.84	106.89	37.88	39.42	76.85	3664.40
	all	52.73	251.92	47.30	22.53	24.97	17.98	3238.73
3	01-05	27.93	161.67	19.96	46.90	48.33	6.46	0.09
	06-09	5.30	296.04	2.88	15.75	16.97	1.80	4284.56
	10-14	2.98	3.54	3.05	1.70	2.99	1.24	3007.97
	15-19	6.65	330.75	7.17	10.40	10.91	4.27	1946.15
	20-24	48.43	245.90	48.41	35.12	35.83	34.61	2846.85
	all	18.80	203.89	16.85	22.23	23.26	10.01	2339.31
5	01-05	51.80	38.33	27.47	4.68	5.76	0.57	0.06
	06-09	5.14	320.47	2.36	2.69	3.37	1.41	3130.59
	10-14	2.23	2.44	2.44	1.61	2.59	0.79	2129.87
	15-19	9.35	223.61	10.75	7.59	7.88	3.04	1445.89
	20-24	44.44	1161.38	180.26	10.89	11.20	8.73	1078.98
	all	23.32	350.44	46.42	5.61	6.28	2.97	1491.52
10	01-05	45.53	12.46	12.47	11.64	12.62	0.05	0.05
	06-09	5.26	454.39	1.79	1.77	2.16	1.01	1967.78
	10-14	1.99	2.51	2.27	2.24	3.00	0.58	1477.10
	15-19	3.88	316.59	15.80	8.56	8.79	2.93	1120.49
	20-24	31.62	434.65	175.75	19.60	19.75	23.51	783.99
	all	18.17	235.36	43.27	9.05	9.56	5.81	1032.47
all	01-05	58.84	153.44	41.50	26.33	28.74	1.91	0.12
	06-09	9.09	272.05	2.69	6.69	7.76	1.75	4325.93
	10-14	2.54	3.04	3.21	1.83	2.96	1.04	2420.13
	15-19	6.67	322.13	9.92	11.93	12.42	3.85	1747.88
	20-24	60.30	553.69	127.83	25.87	26.55	35.93	2093.55
	all	28.26	260.40	38.46	14.86	16.02	9.19	2025.51

Benefits and Perspectives

Cross-Domain Benefits

- achieved substantial performance gains in various optimization domains [25]:
  - Traveling Salesperson Problem [27, 24, 55, 47]
  - Single-Objective Continuous Optimization [5, 21]
  - Propositional Satisfiability (SAT) Problem [53, 54, 33, 29, 30]
  - Traveling Thief Problem [50]
  - Bin-Packing Problem [2]
  - Answer Set Programming (ASP) [19, 29, 30]
  - Quantified Boolean Formula (QBF) Problem [29, 30]
  - AI Planning [44, 45, 8]
  - ...

Automated Algorithm Selection: Survey and Perspectives

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

Holger H. Hoos hh@liacs.nl
Leiden Institute of Advanced Computer Science, Leiden University, 2333 CA Leiden, The Netherlands

Frank Neumann frank.neumann@adelaide.edu.au
Optimisation and Logistics, The University of Adelaide, Adelaide, SA 5005, Australia

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

https://doi.org/10.1162/evco_a_00242

Abstract

It has long been observed that for practically any computational problem that has been intensely studied, different instances are best solved using different algorithms. This is particularly pronounced for computationally hard problems, where in most cases, no single algorithm defines the state of the art; instead, there is a set of algorithms with complementary strengths. This performance complementarity can be exploited in various ways, one of which is based on the idea of selecting, from a set of given algorithms, for each problem instance to be solved the one expected to perform best. The task of automatically selecting an algorithm from a given set is known as the *per-instance algorithm selection problem* and has been intensely studied over the past 15 years, leading to major improvements in the state of the art in solving a growing number of discrete combinatorial problems, including propositional satisfiability and AI planning. Per-instance algorithm selection also shows much promise for boosting performance in solving continuous and mixed discrete/continuous optimisation problems. This survey provides an overview of research in automated algorithm selection, ranging from early and seminal works to recent and promising application areas. Different from earlier work, it covers applications to discrete and continuous problems, and discusses algorithm selection in context with conceptually related approaches, such as algorithm configuration, scheduling, or portfolio selection. Since informative and cheaply computable problem instance features provide the basis for effective per-instance algorithm selection systems, we also provide an overview of such features for discrete and continuous problems. Finally, we provide perspectives on future work in the area and discuss a number of open research challenges.

Keywords

Automated algorithm selection, automated algorithm configuration, combinatorial optimisation, continuous optimisation, machine learning, metalearning, feature-based approaches, exploratory landscape analysis, data streams.

1 Introduction

It has long been observed that for well-studied computational problems for which several high-performance algorithms are available, there is typically no single algorithm

Manuscript received: 17 October 2018; revised: 13 November 2018 and 18 November 2018; accepted: 19 November 2018.
 Evolutionary Computation 27(1): 3–45

Cross-Domain Benefits

- AAS enables to exploit the potential (of portfolios) of state-of-the-art optimization algorithms [6]
- aims at closing the gap between a portfolio's SBS (0%) and VBS (100%)

	classif/ ksvm (0.48)	classif/ randomForest (0.52)	classif/ rpart (0.42)	regr/ lm (0.48)	regr/ randomForest (0.65)	regr/ rpart (0.34)	cluster/ XMeans (0.22)
SAT11–HAND	0.72*	0.62	0.59	0.48	0.65	0.47	0.12
SAT11–INDU	0.10	0.15*	0.11	0.13	0.01	–0.29	0.01
SAT11–RAND	0.76	0.85	0.86	0.83	0.90*	0.82	0.54
SAT12–ALL	0.63	0.70	0.06	0.43	0.72*	0.32	–0.01
SAT12–HAND	0.60	0.70	0.37	0.51	0.75*	0.45	0.08
SAT12–INDU	0.37	0.44	–0.18	0.24	0.47*	–0.01	–0.16
SAT12–RAND	–0.00	0.10	0.34*	0.28	0.25	–0.50	0.10
SAT15–INDU	0.25	0.22	0.21	0.09	0.49*	0.02	–0.10
QBF–2011	0.67	0.79	0.72	0.71	0.88*	0.70	0.43
QBF–2014	0.46	0.53	0.38	0.35	0.79*	0.52	0.07
MAXSAT12–PMS	0.62	0.65	0.62	0.63	0.90*	0.62	0.60
MAXSAT15–PMS–INDU	0.31	0.37	0.27	0.46	0.64*	0.19	0.45
CSP–2010	0.55	0.67	0.83*	0.66	0.77	0.29	0.23
CSP–MZN–2013	0.82	0.79	0.68	0.83	0.91*	0.77	0.16
PROTEUS–2014	0.73	0.73	0.47	0.77	0.81*	0.71	0.54
ASP–POTASSCO	0.39	0.43	0.47	0.55	0.80*	0.53	0.40
PREMAR–ASTAR–2015	0.23	0.16	0.23	0.16	0.29*	0.16	0.22

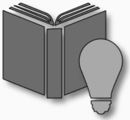
Key Benefits

Profitable Performance

- large speed-up compared to (standalone versions of) state-of-the-art algorithms
- requires less resources / leads to increased profitability

Automated Optimization

- independent of human experts
- high likelihood to select a well-performing algorithm



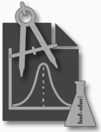
Increased Interpretability

- captures interplay between problem structure and algorithm performance
- key to interpretability of algorithmic search behavior

Research Perspectives

From Text Book to Real-World

- generate more “real-worldish” problem instances
- consider streaming data
- augment data (sensors, weather, traffic, etc.)



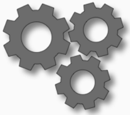
Theoretical Fundamentals of Features

- increase robustness (w.r.t. problem size, scaling, etc.)
- features capable of dealing with stochasticity

Dynamic Algorithm Selection (and Configuration)

- monitor search behavior during the optimization run
- develop features that characterize the search behavior
- switch and configure algorithms dynamically during the run

Research Perspectives



Improve Algorithm Selection Models

- deepen understanding about potential of deep learning
- incorporate automated feature selection strategies

Performance Assessment

- measure the anytime behavior
- multi-objective view (e.g., solution quality vs. runtime)

Apply Concepts to Various TSP-Extensions

- weighted TSP
- Traveling Thief Problem (TTP)
- variants of the Vehicle Routing Problem (VRP)

Thank You!

Any Comments or Questions?!

References

[1] Jason Adair, Gabriela Ochoa, and Katherine Mary Malan. Local Optima Networks for Continuous Fitness Landscapes. In Proceedings of the Genetic and Evolutionary Computation Conference (GECCO) Companion, pages 1407 – 1414. ACM, 2019. doi: 10.1145/3319619.3326852.

[2] Mohamad Alissa, Kevin Sim, and Emma Hart. Algorithm selection using deep learning without feature extraction. In Proceedings of the 21st Annual Conference on Genetic and Evolutionary Computation (GECCO), pages 198 – 206, 2019. doi: 10.1145/3321707.3321845.

[3] David L. Applegate, Robert E. Bixby, Vasek Chvátal, and William J. Cook. The Traveling Salesman Problem: A Computational Study. Princeton University Press, Princeton, NJ, USA, 2007. ISBN 9780691129938. URL http://press.princeton.edu/titles/8451.html.

[4] Thomas Bartz-Beielstein, Carola Doerr, Daan van den Berg, Jakob Bossek, Sowmya Chandrasekaran, Tome Eftimov, Andreas Fischbach, Pascal Kerschke, Manuel López-Ibáñez, Katherine M. Malan, Jason H. Moore, Orzechowski Patry Naujoks, Boris, Vanessa Volz, Markus Wagner, and Thomas Weise. Benchmarking in Optimization: Best Practice and Open Issues. arXiv preprint, abs/2007.03488, 2020. URL https://arxiv.org/abs/2007.03488.

[5] Bernd Bischl, Olaf Mersmann, Heike Trautmann, and Mike Preuss. Algorithm Selection Based on Exploratory Landscape Analysis and Cost-Sensitive Learning. In Proceedings of the 14th Annual Conference on Genetic and Evolutionary Computation (GECCO), pages 313 – 320. ACM, July 2012. ISBN 978-1-4503-0557-0. doi: 10.1145/2330163.2330209. URL http://dl.acm.org/citation.cfm?doid=2330163.2330209.

[6] Bernd Bischl, Pascal Kerschke, Lars Kotthoff, Thomas Marius Lindauer, Yuri Malitsky, Alexandre Fréchette, Holger H. Hoos, Frank Hutter, Kevin Leyton-Brown, Kevin Tierney, and Joaquin Vanschoren. ASlib: A Benchmark Library for Algorithm Selection. Artificial Intelligence (AIJ), 237:41 – 58, 2016. ISSN 0004-3702. doi: 10.1016/j.artint.2016.04.003. URL http://www.sciencedirect.com/science/article/pii/S0004370216300388.

[7] Jakob Bossek, Pascal Kerschke, Aneta Neumann, Markus Wagner, Frank Neumann, and Heike Trautmann. Evolving diverse tsp instances by means of novel and creative mutation operators. In Proceedings of the 15th ACM/SIGEVO Conference on Foundations of Genetic Algorithms (FOGA), pages 58 – 71. ACM, 2019. doi: 10.1145/3299904.3340307.

[8] Chris Cameron, Holger H. Hoos, Kevin Leyton-Brown, and Frank Hutter. OASC-2017: *Zilla Submission. In Thomas Marius Lindauer, Jan Nicolaas van Rijn, and Lars Kotthoff, editors, Proceedings of the Open Algorithm Selection Challenge, volume 79 of Proceedings of Machine Learning Research, pages 15–18, Brussels, Belgium, September 2017. URL http://proceedings.mlr.press/v79/cameron17a.html.

[9] Bilel Derbel, Arnaud Liefooghe, Sébastien Verel, Hernan Aguirre, and Kiyoshi Tanaka. New Features for Continuous Exploratory Landscape Analysis Based on the SOO Tree. In Proceedings of the 15th ACM/SIGEVO Conference on Foundations of Genetic Algorithms (FOGA XV), pages 72 – 86. ACM, 2019. doi: 10.1145/3299904.3340308.

[10] Carola Doerr, Hao Wang, Furong Ye, Sander van Rijn, and Thomas Bäck. Iohprofiler: A benchmarking and profiling tool for iterative optimization heuristics. arXiv e-prints:1810.05281, Oct 2018. URL https://arxiv.org/abs/1810.05281.

[11] Jérémie Dubois-Lacoste, Holger H. Hoos, and Thomas Stützle. On the Empirical Scaling Behaviour of State-of-the-art Local Search Algorithms for the Euclidean TSP. In Proceedings of the 17th Annual Conference on Genetic and Evolutionary Computation (GECCO), pages 377 – 384, New York, NY, USA, 2015. ACM. ISBN 978-1-4503-3472-3. doi: 10.1145/2739480.2754747. URL http://dl.acm.org/citation.cfm?id=2754747.

[12] Nikolaus Hansen, Steffen Finck, Raymond Ros, and Anne Auger. Real-Parameter Black-Box Optimization Benchmarking 2009: Noiseless Functions Definitions. Technical Report RR-6829, INRIA, 2009. URL https://hal.inria.fr/inria-00362633/document.

[13] Nikolaus Hansen, Anne Auger, Olaf Mersmann, Tea Tušar, and Dimo Brockhoff. COCO: A Platform for Comparing Continuous Optimizers in a Black-Box Setting. arXiv preprint, abs/1603.08785v3, August 2016. URL http://arxiv.org/abs/1603.08785v3.

[14] Nikolaus Hansen, Anne Auger, Raymond Ros, Olaf Mersmann, Tea Tušar, and Dimo Brockhoff. Coco: A platform for comparing continuous optimizers in a black-box setting. Optimization Methods and Software, 36(1):114 – 144, 2021. doi: 10.1080/10556788.2020.1808977.

[15] Christian Hanster and Pascal Kerschke. flaccogui: Exploratory Landscape Analysis for Everyone. In Proceedings of the 19th Annual Conference on Genetic and Evolutionary Computation (GECCO) Companion, pages 1215 – 1222. ACM, July 2017. doi: 10.1145/3067695.3082477. URL http://dl.acm.org/citation.cfm?doid=3067695.3082477.

[16] Jonathan Heins, Jakob Bossek, Janina Pohl, Moritz Seiler, Heike Trautmann, and Pascal Kerschke. On the potential of normalized tsp features for automated algorithm selection. In Proceedings of the 16th ACM/SIGEVO Conference on Foundations of Genetic Algorithms (FOGA), pages 1 – 15. ACM, 2021. doi: 10.1145/3450218.3477308.

[17] Jonathan Heins, Jakob Bossek, Janina Pohl, Moritz Seiler, Heike Trautmann, and Pascal Kerschke. A study on the effects of normalized tsp features for automated algorithm selection. Theoretical Computer Science, 940(Part B):123 – 145, 2022. doi: 10.1016/j.tcs.2022.10.019.

[18] Keld Helsgaun. General k-opt submoves for the Lin-Kernighan TSP heuristic. Mathematical Programming Computation, 1 (2-3):119 – 163, 2009. doi: 10.1007/s12532-009-0004-6. URL https://link.springer.com/article/10.1007/s12532-009-0004-6.

[19] Holger H. Hoos, Thomas Marius Lindauer, and Torsten Schaub. claspfolio 2: Advances in Algorithm Selection for Answer Set Programming. Theory and Practice of Logic Programming (TPLP), 14(4-5):569 – 585, 2014. doi: 10.1017/S1471068414000210. URL https://www.cambridge.org/core/journals/theory-and-practice-of-logic-programming/article/claspfolio-2-advances-in-algorithm-selection-for-answer-set-programming/24B21F6FB773164716AB63C6099C7755.

[20] Frank Hutter, Lin Xu, Holger H. Hoos, and Kevin Leyton-Brown. Algorithm Runtime Prediction: Methods & Evaluation. Artificial Intelligence (AIJ), 206:79 – 111, 2014. ISSN 0004-3702. URL http://www.sciencedirect.com/science/article/pii/S0004370213001082.

[21] Pascal Kerschke and Heike Trautmann. Automated Algorithm Selection on Continuous Black-Box Problems By Combining Exploratory Landscape Analysis and Machine Learning. Evolutionary Computation (ECJ), 27:99 – 127, 2019. doi: 10.1162/evco_a_00236.

[22] Pascal Kerschke and Heike Trautmann. Comprehensive Feature-Based Landscape Analysis of Continuous and Constrained Optimization Problems Using the R-package flacco. In Applications in Statistical Computing, pages 93 – 123. Springer, 2019. doi: 10.1007/978-3-030-25147-5_7.

[23] Pascal Kerschke, Mike Preuss, Carlos Ignacio Hernández Castellanos, Oliver Schütze, Jian-Qiao Sun, Christian Grimme, Günter Rudolph, Bernd Bischl, and Heike Trautmann. Cell Mapping Techniques for Exploratory Landscape Analysis. In EVOLVE – A Bridge between Probability, Set Oriented Numerics, and Evolutionary Computation V, pages 115 – 131. Springer, July 2014. ISBN 978-3-319-07493-1. doi: 10.1007/978-3-319-07494-8_9. URL https://link.springer.com/chapter/10.1007/978-3-319-07494-8_9.

[24] Pascal Kerschke, Lars Kotthoff, Jakob Bossek, Holger H. Hoos, and Heike Trautmann. Leveraging TSP Solver Complementarity through Machine Learning. Evolutionary Computation (ECJ), 26:597 – 620, 2018. doi: 10.1162/evco_a_00215.

[25] Pascal Kerschke, Holger H. Hoos, Frank Neumann, and Heike Trautmann. Automated Algorithm Selection: Survey and Perspectives. Evolutionary Computation (ECJ), 27:3 – 45, 2019. doi: 10.1162/evco_a_00242.

[26] Lars Kotthoff. Algorithm Selection for Combinatorial Search Problems: A Survey. AI Magazine, 35(3):48 – 60, 2014. ISSN 0738-4602. doi: 10.1609/aimag.v35i3.2460. URL https://aaai.org/ojs/index.php/aimagazine/article/view/2460.

[27] Lars Kotthoff, Pascal Kerschke, Holger H. Hoos, and Heike Trautmann. Improving the State of the Art in Inexact TSP Solving Using Per-Instance Algorithm Selection. In Clarisse Dhaenens, Laetitia Jourdan, and Marie-Eléonore Marmion, editors, Proceedings of the 9th International Conference on Learning and Intelligent Optimization (LION), volume 8994 of Lecture Notes in Computer Science (LNCS), pages 202 – 217. Springer, January 2015. ISBN 978-3-319-19084-6. doi: 10.1007/978-3-319-19084-6_18. URL https://link.springer.com/chapter/10.1007%2F978-3-319-19084-6_18.

[28] Michel Lang, Martin Binder, Jakob Richter, Patrick Schratz, Florian Pfisterer, Stefan Coors, Quay Au, Giuseppe Casalicchio, Lars Kotthoff, and Bernd Bischl. mlr3: A modern object-oriented machine learning framework in R. Journal of Open Source Software, dec 2019. doi: 10.21105/joss.01903. URL https://joss.theoj.org/papers/10.21105/joss.01903.

[29] Thomas Marius Lindauer, Holger H. Hoos, Frank Hutter, and Thorsten Schaub. AutoFolio: An Automatically Configured Algorithm Selector. Journal of Artificial Intelligence Research (JAIR), 53:745 – 778, 2015.

[30] Thomas Marius Lindauer, Holger H. Hoos, Frank Hutter, and Thorsten Schaub. AutoFolio: An Automatically Configured Algorithm Selector (Extended Abstract). In Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI), pages 5025 – 5029, August 2017. doi: 10.24963/ijcai.2017/715. URL https://www.ijcai.org/proceedings/2017/715.

[31] Ilya Loshchilov, Marc Schoenauer, and Michèle Sebag. Bi-Population CMA-ES Algorithms with Surrogate Models and Line Searches. In Proceedings of the 15th Annual Conference on Genetic and Evolutionary Computation (GECCO), pages 1177 – 1184. ACM, July 2013. doi: 10.1145/2464576.2482696. URL http://dl.acm.org/citation.cfm?doid=2464576.2482696.

[32] Monte Lunacek and L. Darrell Whitley. The Dispersion Metric and the CMA Evolution Strategy. In Proceedings of the 8th Annual Conference on Genetic and Evolutionary Computation (GECCO), pages 477 – 484. ACM, 2006. doi: 10.1145/1143997.1144085.

[33] Yuri Malitsky, Ashish Sabharwal, Horst Samulowitz, and Meinolf Sellmann. Boosting Sequential Solver Portfolios: Knowledge Sharing and Accuracy Prediction. In Giuseppe Nicosia and Panos Pardalos, editors, Proceedings of the 7th International Conference on Learning and Intelligent Optimization (LION), volume 7997 of Lecture Notes in Computer Science (LNCS), pages 153 – 167. Springer, January 2013. doi: 10.1007/978-3-642-44973-4_17. URL https://link.springer.com/chapter/10.1007/978-3-642-44973-4_17.

[34] Olaf Mersmann, Bernd Bischl, Heike Trautmann, Mike Preuss, Claus Weihs, and Günter Rudolph. Exploratory Landscape Analysis. In Proceedings of the 13th Annual Conference on Genetic and Evolutionary Computation (GECCO), pages 829 – 836. ACM, 2011. ISBN 978-1-4503-0557-0. doi: 10.1145/2001576.2001690.

[35] Olaf Mersmann, Bernd Bischl, Heike Trautmann, Markus Wagner, Jakob Bossek, and Frank Neumann. A Novel Feature-Based Approach to Characterize Algorithm Performance for the Traveling Salesperson Problem. Annals of Mathematics and Artificial Intelligence, 69:151 – 182, October 2013. doi: 10.1007/s10472-013-9341-2. URL https://link.springer.com/article/10.1007/s10472-013-9341-2.

[36] Mario Andrés Muñoz Acosta, Michael Kirley, and Saman K. Halgamuge. Exploratory Landscape Analysis of Continuous Space Optimization Problems Using Information Content. IEEE Transactions on Evolutionary Computation (TEVC), 19(1):74 – 87, 2015. ISSN 1089-778X. doi: 10.1109/TEVC.2014.2302006. URL http://ieeexplore.ieee.org/document/6719480/.

[37] Yuichi Nagata and Shigenobu Kobayashi. A Powerful Genetic Algorithm Using Edge Assembly Crossover for the Traveling Salesman Problem. INFORMS Journal on Computing, 25(2):346 – 363, 2013. doi: 10.1287/ijoc.1120.0506.

[38] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. Journal of Machine Learning Research, 12:2825–2830, 2011.

[39] Josef Pihera and Nysret Musliu. Application of Machine Learning to Algorithm Selection for TSP. In Proceedings of the IEEE 26th International Conference on Tools with Artificial Intelligence (ICTAI). IEEE, November 2014. doi: 10.1109/ICTAI.2014.18. URL https://www.dbai.tuwien.ac.at/staff/musliu/art_ictai_cam.pdf.

[40] Raphael Prager, Heike Trautmann, and Pascal Kerschke. (p)flacco: Exploratory Landscape Analysis of Single-Objective Continuous Black-Box Problems in R and Python. IEEE Transactions on Evolutionary Computation (TEVC), 2021. Special Issue on Benchmarking Sampling-Based Optimization Heuristics: Methodology and Software (BENCH), submitted.

[41] Raphael Patrick Prager. Pflacco: Feature-Based Landscape Analysis of Continuous and Constrained Optimization Problems in Python. https://github.com/Reiyan/pflacco, 2022.

[42] Raphael Patrick Prager, Heike Trautmann, Hao Wang, Thomas H. W. Bäck, and Pascal Kerschke. Per-Instance Configuration of the Modularized CMA-ES by Means of Classifier Chains and Exploratory Landscape Analysis. Proceedings of the 2020 IEEE Symposium Series on Computational Intelligence (SSCI), pages 996 – 1003, 2020. doi: 10.1109/SSCI47803.2020.9308510.

[43] John Rischard Rice. The Algorithm Selection Problem. Advances in Computers, 15:65 – 118, 1976. doi: 10.1016/S0065-2458(08)60520-3. URL http://www.sciencedirect.com/science/article/pii/S0065245808605203.

[44] Mattia Rizzini, Chris Fawcett, Mauro Vallati, Alfonso E Gerevini, and Holger H. Hoos. Portfolio Methods for Optimal Planning: an Empirical Analysis. In Proceedings of the IEEE 27th International Conference on Tools with Artificial Intelligence (ICTAI), pages 494 – 501. IEEE, November 2015. doi: 10.1109/ICTAI.2015.79. URL http://ieeexplore.ieee.org/document/7372175/.

[45] Mattia Rizzini, Chris Fawcett, Mauro Vallati, Alfonso Emilio Gerevini, and Holger H. Hoos. Static and Dynamic Portfolio Methods for Optimal Planning: An Empirical Analysis. International Journal on Artificial Intelligence Tools, 26(01):1 – 27, 2017. doi: 10.1142/S0218213017600065. URL https://www.worldscientific.com/doi/abs/10.1142/S0218213017600065.

[46] Lennart Schäpermeier, Christian Grimme, and Pascal Kerschke. Plotting Impossible? Surveying Visualization Methods for Continuous Multi-Objective Benchmark Problems. IEEE Transactions on Evolutionary Computation, 26(6):1306 – 1320, 2022. doi: 10.1109/TEVC.2022.3214894. URL https://doi.org/10.1109/TEVC.2022.3214894.

[47] Moritz Seiler, Janina Pohl, Jakob Bossek, Pascal Kerschke, and Heike Trautmann. Deep learning as a competitive feature-free approach for automated algorithm selection on the traveling salesperson problem. In Proceedings of the 16th International Conference on Parallel Problem Solving from Nature (PPSN), pages 48 – 64. Springer, 2020. doi: 10.1007/978-3-030-58112-1_4.

[48] Kate Amanda Smith-Miles. Cross-Disciplinary Perspectives on Meta-Learning for Algorithm Selection. ACM Computing Surveys (CSUR), 41:1 – 25, January 2009. doi: 10.1145/1456650.1456656. URL https://dl.acm.org/citation.cfm?doid=1456650.1456656.

[49] Sander van Rijn, Hao Wang, Bas van Stein, and Thomas Bäck. Algorithm Configuration Data Mining For CMA Evolution Strategies. In Proceedings of the 19th Annual Conference on Genetic and Evolutionary Computation (GECCO), pages 737 – 744. ACM, July 2017. doi: 10.1145/3071178.3071205. URL https://dl.acm.org/citation.cfm?id=3071178.3071205.

[50] Markus Wagner, Thomas Marius Lindauer, Mustafa Mısır, Samadhi Nallaperuma, and Frank Hutter. A Case Study of Algorithm Selection for the Traveling Thief Problem. Journal of Heuristics, 24:295 – 320, 2018. doi: 10.1007/s10732-017-9328-y. URL https://link.springer.com/article/10.1007/s10732-017-9328-y.

[51] David Hilton Wolpert and William G. Macready. No Free Lunch Theorems for Optimization. IEEE Transactions on Evolutionary Computation (TEVC), 1(1):67 – 82, April 1997. doi: 10.1109/4235.585893. URL http://ieeexplore.ieee.org/document/585893/.

[52] Xiao-Feng Xie and Jiming Liu. Multiagent Optimization System for Solving the Traveling Salesman Problem (TSP). IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics, 39(2):489 – 502, April 2009. doi: 10.1109/TSMCB.2008.2006910. URL http://ieeexplore.ieee.org/document/4717264/.

[53] Lin Xu, Frank Hutter, Holger H. Hoos, and Kevin Leyton-Brown. SATzilla: Portfolio-Based Algorithm Selection for SAT. Journal of Artificial Intelligence Resesearch (JAIR), 32:565 – 606, 2008. URL http://www.cs.ubc.ca/labs/beta/Projects/SATzilla/.

[54] Lin Xu, Frank Hutter, Holger H. Hoos, and Kevin Leyton-Brown. Evaluating Component Solver Contributions to Portfolio-Based Algorithm Selectors. In Alessandro Cimatti and Roberto Sebastiani, editors, Proceedings of the 15th International Conference on Theory and Applications of Satisfiability Testing (SAT), volume 7317 of Lecture Notes in Computer Science (LNCS), pages 228 – 241. Springer, June 2012. URL https://link.springer.com/chapter/10.1007/978-3-642-31612-8_18.

[55] Kangfei Zhao, Shengcai Liu, Yu Rong, and Jeffrey Xu Yu. Leveraging tsp solver complementarity via deep learning. arXiv preprints, abs/2006.00715, 6 2020. URL https://arxiv.org/abs/2006.00715.