Advanced Use of Automatic Algorithm Configuration

Jeroen Rook

j.g.rook@utwente.nl
http://jeroenrook.nl

University of Twente, NL

Manuel López-Ibáñez

manuel.lopez-ibanez@manchester.ac.uk
http://lopez-ibanez.eu

University of Manchester, UK

Heike Trautmann

heike.trautmann@uni-paderborn.de
http://maleo-research.de

Paderborn University, DE

PPSN 2024, 15 September 2024, Hagenberg im Mühlkreis


LOVELY AND BRIGHT SPACE!

Tutorial Objectives

- Go beyond the *classical* AAC scenario
- Special focus on dealing with multiple objectives
  - At the algorithm level
  - At the performance level

Profile

You are (somewhat) familiar with ...

- Automated Algorithm Configuration
- Multi-objective optimization

You understand the importance of including AAC in research involving benchmarking. i.e. anywhere where you compare the performance between algorithms.

Profile

You are (somewhat) familiar with ...

- Automated Algorithm Configuration
- Multi-objective optimization

You understand the importance of including AAC in research involving benchmarking. i.e. anywhere where you compare the performance between algorithms.

No perfect fit? No worries!

Agenda

Part I: Crash course on
AAC and Multi-objective optimisation

Part II: AAC for Multi-Objective Optimization Algorithms

Part III: AAC for Improving Anytime Behaviour

Part IV: AAC for Multiple Performance Objectives

Part V: Wrap-up

Part I

Crash course on AAC and Multi-objective optimisation

AAC formal definition

Find a configuration for an algorithm
that optimises the overall performance for a specific task.

AAC formal definition

Find a configuration for an algorithm
that optimises the overall performance for a specific task.

Formulated as optimisation problem:

$$\theta^* = \arg\max_{\theta \in \Theta} \mathbb{E}_{\pi \sim \mathcal{D}} \; p(A_\theta, \; \pi)$$

- Θ Configuration space
- A Algorithm
- $\mathcal{I}$ Problem domain
- $\mathcal{D}$ Distribution over problem instances with domain $\mathcal{I}$
- p Performance measure $p : \Theta \times \mathcal{I} \to \mathbb{R}$

AAC formal definition

Find a configuration for an algorithm
that optimises the overall performance for a specific task.

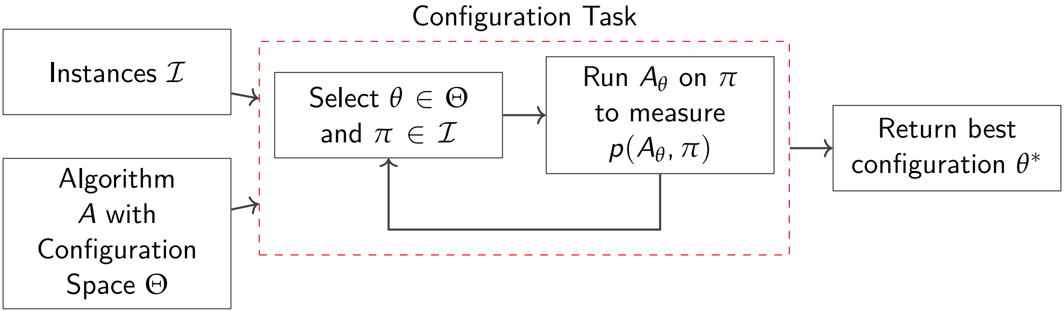
Formulated as optimisation problem:

$$\theta^* = \arg\max_{\theta \in \Theta} \mathbb{E}_{\pi \sim \mathcal{D}} \; p(A_\theta, \; \pi)$$

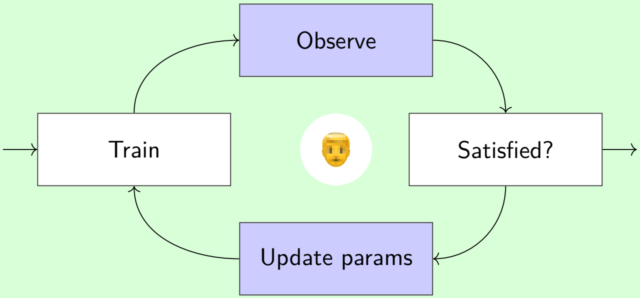
- Θ Configuration space
- A Algorithm
- $\mathcal{I}$ Problem domain
- $\mathcal{D}$ Distribution over problem instances with domain $\mathcal{I}$
- p Performance measure $p : \Theta \times \mathcal{I} \to \mathbb{R}$

$\mathcal{I}$ is usually represented by a set of instances (N)

AAC diagram


Configuration Task
Instances $\mathcal{I}$
Algorithm A with Configuration Space Θ
Select $\theta \in \Theta$ and $\pi \in \mathcal{I}$
Run A_θ on π to measure $p(A_\theta, \pi)$
Return best configuration θ^*

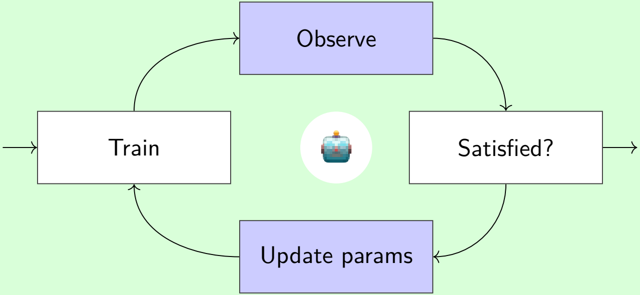
Algorithm Configuration



Human

- Slow
- Biased
- Untrackable

Automated Algorithm Configuration



Human

- Slow
- Biased
- Untrackable

Machine

- Fast
- *Unbiased*
- Systematic

AAC - Configuration space Θ

Parameter configuration space (PCS) [Hutter & Ramage, 2015]

- Name, type, range & default `a integer [0,255] [8]`
- Conditional parameters `b | c in {foo}`
- Forbidden combinations `a=0, c=foo`

Example for `sklearn.models.RandomForest`:

```
bootstrap categorical {True, False} [True]
criterion categorical {gini, entropy, log_loss} [gini]
max_depth_type categorical {None, int} [None]
max_depth integer [1, 100] [10]
max_depth | max_depth_type == int
```

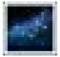
$$\Theta = \{(\mathit{True}, \mathit{gini}, \mathit{None}, -), (\mathit{True}, \mathit{gini}, \mathit{int}, 1), \dots\}$$

$$|\Theta| = 606$$

AAC - Parameter space Θ

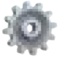
```
bootstrap categorical {True, False} [True]
criterion categorical {gini, entropy, log_loss} [gini]
max_depth_type categorical {None, int} [None]
max_features_type categorical {special, float} [special]
max_leaf_nodes_type categorical {None, int} [None]
min_impurity_decrease real [0.0, 0.5] [0.0]
min_samples_leaf integer [1, 100] [1]
min_samples_split integer [1, 100] [2]
min_weight_fraction_leaf real [0.0, 0.5] [0.0]
n_estimators integer [1, 500] [100]
max_depth integer [1, 100] [10]
max_features_float real [0.0, 1.0] [0.5]
max_features_special categorical {sqrt, log2, None} [sqrt]
max_leaf_nodes integer [1, 1000] [100]
max_samples_type categorical {None, float} [None]
oob_score categorical {True, False} [True]
max_samples real [0.05, 0.95] [0.8]

max_samples_type | bootstrap == True
oob_score | bootstrap == True
max_depth | max_depth_type == int
max_features_float | max_features_type == float
max_features_special | max_features_type == special
max_leaf_nodes | max_leaf_nodes_type == int
max_samples | max_samples_type == float
```

Challenges – Large search spaces



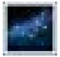
Planets in the universe

Atoms on earth



Unique configurations

Challenges – Large search spaces

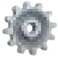


Planets in the universe

$\approx \ 10^{23}$

Atoms on earth

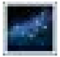
$\approx \ 10^{50}$



Unique configurations

$\approx \ 10^{24}$

Challenges – Large search spaces

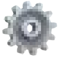


Planets in the universe

$\approx \ 10^{23}$

Atoms on earth

$\approx \ 10^{50}$

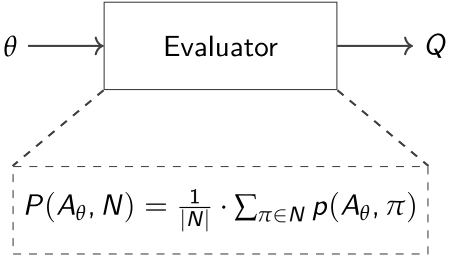


Unique configurations

$\approx \ 10^{24}$

SAT solver *lingeling* has 10^{947} distinct configurations

Challenges – Expensive evaluations



$$P(A_\theta, N) = \frac{1}{|N|} \cdot \sum_{\pi \in N} p(A_\theta, \pi)$$

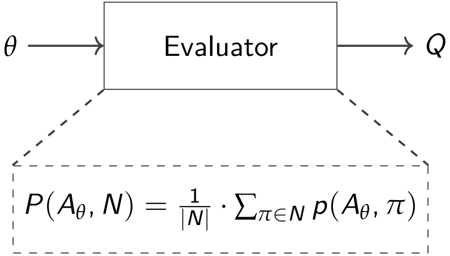
Challenges – Expensive evaluations

$$\theta \longrightarrow \boxed{\text{Evaluator}} \longrightarrow Q$$

$$P(A_\theta, N) = \frac{1}{|N|} \cdot \sum_{\pi \in N} p(A_\theta, \pi)$$

Example:
100 instances, $\approx 30s$ to run $\rightarrow 3000s \ \approx$ 50 minutes

Challenges – Expensive evaluations



Example:

100 instances, $\approx 30s$ to run $\rightarrow 3000s \approx 50$ minutes

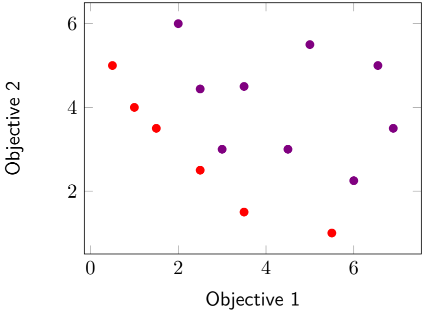
606 configurations $\cdot$ 50 minutes $\rightarrow$ 21.04 days

AAC challenges

- Large, mixed-type and nested search spaces
- Expensive evaluations
- Many 'bad' configurations compared to the default parameters

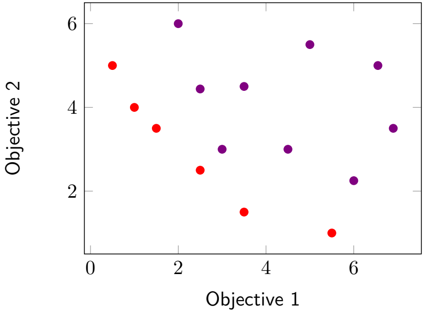
Multi-objective Optimization

- Optimize for multiple *conflicting* objectives.
- Obtain solution set that is the trade-off between the objectives, i.e. Pareto Set.
- No other solution should (Pareto) dominate elements in the solution set.
- Projection of solution set in decision space is Pareto front.
- With EMOAs we approximate the Pareto set.



Multi-objective Optimization

- Optimize for multiple *conflicting* objectives.
- Obtain solution set that is the trade-off between the objectives, i.e. Pareto Set.
- No other solution should (Pareto) dominate elements in the solution set.
- Projection of solution set in decision space is Pareto front.
- With EMOAs we approximate the Pareto set.
- How to compare sets against other sets?



Performance indicators

- Many indicators to measure quality:
  - Hypervolume / $\mathcal{S}$-metric
  - IGD
  - IGD$^+$
  - ϵ-indicator
  - R2-indicator
  - Averaged Hausdorff distance (Δ_q)
  - Riesz S-energy
  - Cone-based hypervolume
  - ...
- $>$ 100 indicators recorded. [Zitzler et al., 2003; Knowles et al., 2006; Audet et al., 2021]
- Aggregating indicators over various problem instances not always trivial.
- Need for reference sets, vectors or points.
- Understand how indicators trade each other off /
  Find configurations that compromise well on the selected indicators.

AAC comes in various forms and names

- Hyper-parameter optimization
- Hyper-heuristics
- Algorithm tuning
- Meta-optimization
- ...

Offline configuration vs. Online control

Offline tuning / Algorithm configuration

- Learn best configuration before *solving* the real problem instance
- Configuration done on training problem instances
- Performance measured over test ($\neq$ training) instances

Offline configuration vs. Online control

Offline tuning / Algorithm configuration

- Learn best configuration before *solving* the real problem instance
- Configuration done on training problem instances
- Performance measured over test ($\neq$ training) instances

Online tuning / Parameter control / Reactive search

- Learn best configuration *while* solving each instance
- No training phase
- Very popular in continuous optimization
- Ultimate goal: parameter-free algorithms

Offline configuration vs. Online control

Offline tuning / Algorithm configuration

- Learn best configuration before *solving* the real problem instance
- Configuration done on training problem instances
- Performance measured over test ($\neq$ training) instances

Online tuning / Parameter control / Reactive search

- Learn best configuration *while* solving each instance
- No training phase
- Very popular in continuous optimization
- Ultimate goal: parameter-free algorithms

All online methods have parameters that are configured offline

MO tuning vs. Tuning MO algorithms

[Bezerra et al., 2020b]

Multi-objective AAC

- Multiple metrics to evaluate an algorithm configuration
- AAC produces mutually nondominated *set* of configurations

MO tuning vs. Tuning MO algorithms [Bezerra et al., 2020b]

Multi-objective AAC

- Multiple metrics to evaluate an algorithm configuration
- AAC produces mutually nondominated *set* of configurations

AAC for multi-objective algorithms

- Running a configuration outputs a *set* of mutually nondominated solutions (and/or *anytime* behavior)
- Unary quality metrics (Hypervolume, epsilon, IGD+) evaluate the output [Zitzler et al., 2003]
- Uses *single-objective* AAC methods and produces a single best

  [López-Ibáñez & Stützle, 2012; Bezerra et al., 2016; Nebro et al., 2019; Bezerra et al., 2020a; Rook et al., 2022]

MO tuning vs. Tuning MO algorithms [Bezerra et al., 2020b]

Multi-objective AAC

- Multiple metrics to evaluate an algorithm configuration
- AAC produces mutually nondominated *set* of configurations

AAC for multi-objective algorithms

- Running a configuration outputs a *set* of mutually nondominated solutions (and/or *anytime* behavior)
- Unary quality metrics (Hypervolume, epsilon, IGD+) evaluate the output [Zitzler et al., 2003]
- Uses *single-objective* AAC methods and produces a single best [López-Ibáñez & Stützle, 2012; Bezerra et al., 2016; Nebro et al., 2019; Bezerra et al., 2020a; Rook et al., 2022]

Multi-objective AAC of multi-objective algorithms is also possible!

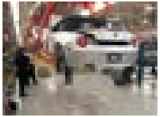
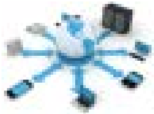
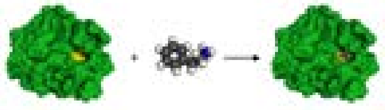
Part II

AAC for Multi-Objective Optimization Algorithms

AutoMOEA

Multi-objective Evolutionary Algorithms

- +30 years of research
- Most researched MO metaheuristic
- Real-world applications in many domains

- Numerous high-quality libraries/frameworks:
  jMetal, PyGMO/PaGMO, PyMOO, PlatEMO, ...

MOEAs: Which one?

- MOGA [Fonseca & Fleming, 1993]
- PAES [Knowles & Corne, 2000]
- NSGA-II [Deb et al., 2002]
- SPEA2 [Zitzler et al., 2002]
- IBEA [Zitzler & Künzli, 2004]
- SMS-EMOA [Beume et al., 2007]
- MO-CMA-ES [Igel et al., 2007]
- MOEA/D [Li & Zhang, 2009]
- HypE [Bader & Zitzler, 2011]
- NSGA-III [Deb & Jain, 2014]

MOEAs: Which one?

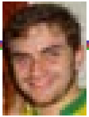
- MOGA [Fonseca & Fleming, 1993]
- PAES [Knowles & Corne, 2000]
- NSGA-II [Deb et al., 2002]
- SPEA2 [Zitzler et al., 2002]
- IBEA [Zitzler & Künzli, 2004]
- SMS-EMOA [Beume et al., 2007]
- MO-CMA-ES [Igel et al., 2007]
- MOEA/D [Li & Zhang, 2009]
- HypE [Bader & Zitzler, 2011]
- NSGA-III [Deb & Jain, 2014]
- GDE3 [Kukkonen & Lampinen, 2005]
- DEMO [Robič & Filipič, 2005]
- DEMO^{SP2}, DEMO^{IB} [Tušar & Filipič, 2007]
- Indicator-based Differential Evolution [Tagawa et al., 2011]

MOEAs: Which one?

- MOGA [Fonseca & Fleming, 1993]
- PAES [Knowles & Corne, 2000]
- NSGA-II [Deb et al., 2002]
- SPEA2 [Zitzler et al., 2002]
- IBEA [Zitzler & Künzli, 2004]
- SMS-EMOA [Beume et al., 2007]
- MO-CMA-ES [Igel et al., 2007]
- MOEA/D [Li & Zhang, 2009]
- HypE [Bader & Zitzler, 2011]
- NSGA-III [Deb & Jain, 2014]
- GDE3 [Kukkonen & Lampinen, 2005]
- DEMO [Robič & Filipič, 2005]
- DEMO^{SP2}, DEMO^{IB} [Tušar & Filipič, 2007]
- Indicator-based Differential Evolution [Tagawa et al., 2011]
- Genetic Diversity Evolutionary Algorithm (GDEA)
- Δ_p-Differential Evolution (DDE)
- neighbourhood exploring evolution strategy (NEES)
- OPTIMOGA
- Biogeography-based multi-objective evolutionary algorithm (BBMOEA)

AutoMOEA

- ✔ Replicate as many well-known MOEAs as possible from the same *template*
- ✔ The template has a number of configurable algorithmic *components*
- ✔ Each component can be configured by choosing one *option* from various alternatives
- ✔ Aim to maximise the number of different configurations that are valid MOEAs



Leonardo C. T. Bezerra, Manuel López-Ibáñez, and Thomas Stützle.
Automatic Component-Wise Design of Multi-Objective Evolutionary Algorithms.
IEEE Transactions on Evolutionary Computation, 2016.

Leonardo C. T. Bezerra, Manuel López-Ibáñez, and Thomas Stützle. **Automatically Designing State-of-the-Art Multi- and Many-Objective Evolutionary Algorithms.**
Evolutionary Computation, 2020.

AutoMOEA: A MOEA template

```
pop := Initialization ()
if typeof(archive) != none then
    archive :=pop
repeat
    pool := BuildMatingPool (pop)
    pop_new := Variation (pool)
    pop_new := Evaluation (pop_new)
    pop     := Replacement (pop, pop_new)
    if typeof(archive) == bounded then
        archive := Archiving (archive, pop_new)
    else if typeof(archive) == unbounded then
        archive := archive ∪ pop
until termination criteria met
if typeof(archive) == none then
    return pop
else
    return archive
```

AutoMOEA: Main components

Component	Parameters
`BuildMatingPool`	$\langle$ `Preference`$_{Mat}$, `Selection` $\rangle$
`Replacement`	$\langle$ `Preference`$_{Rep}$, `Removal` $\rangle$
`Archiving`	$\langle$ `Preference`$_{Ext}$, `Removal`$_{Ext}$ $\rangle$
`Preference`	$\langle$ `Fitness`, `Diversity` $\rangle$

AutoMOEA: Main components

Component	Parameters
`BuildMatingPool`	$\langle$ `Preference`$_{Mat}$, `Selection` $\rangle$
`Replacement`	$\langle$ `Preference`$_{Rep}$, `Removal` $\rangle$
`Archiving`	$\langle$ `Preference`$_{Ext}$, `Removal`$_{Ext}$ $\rangle$
`Preference`	$\langle$ `Fitness`, `Diversity` $\rangle$

Algorithm	Fitness	Diversity
NSGA-II	dominance depth	crowding distance
SPEA2	dom. strength	kNN
IBEA	binary indicator	
HypE	I_H^h	
SMS-EMOA	dom. depth-rank	I_H^1

AutoMOEA: Redesign of main components [Zitzler et al., 2010]

Component	Parameters
`BuildMatingPool`	⟨ `Preference`$_{Mat}$, `Selection` ⟩
`Replacement`	⟨ `Preference`$_{Rep}$, `Removal` ⟩
`Archiving`	⟨ `Preference`$_{Ext}$, `Removal`$_{Ext}$ ⟩
`Preference`	~~⟨ `Fitness`, `Diversity` ⟩~~
	⟨ `Set-partitioning`, `Quality`, `Diversity` ⟩

Component	Parameters
`BuildMatingPool`	⟨ `Preference`$_{Mat}$, `Selection` ⟩
`Replacement`	⟨ `Preference`$_{Rep}$, `Removal` ⟩
`Archiving`	⟨ `Preference`$_{Ext}$, `Removal`$_{Ext}$ ⟩
`Preference`	~~⟨ `Fitness`, `Diversity` ⟩~~
	⟨ `Set-partitioning`, `Quality`, `Diversity` ⟩

	`BuildMatingPool`			`Replacement`		
	`SetPart`	`Quality`	`Diversity`	`SetPart`	`Quality`	`Diversity`
MOGA	dom. rank	—	niche-sharing	—	—	—
NSGA-II	dom. depth	—	crowding dist.	dom. depth	—	crowding dist.
SPEA2	dom. strength	—	kNN	dom. strength	—	kNN
IBEA	—	binary indicator	—	—	binary ind.	—
HypE	—	I_H^h	—	dom. depth	I_H^h	—
SMS-EMOA	—	—	—	dom. depth-rank	I_H^1	—

AutoMOEA: Automatic Design

Automatic configuration (**irace**)
$+$ Flexible algorithmic framework (AutoMOEA)
$=$ Automatic design of state-of-the-art MOEAs

AutoMOEA: Automatic Design

Automatic configuration (**irace**)
$+$ Flexible algorithmic framework (AutoMOEA)
$=$ Automatic design of state-of-the-art MOEAs

	`BuildMatingPool`			`Replacement`		
	`SetPart`	`Quality`	`Diversity`	`SetPart`	`Quality`	`Diversity`
MOGA	rank	—	niche-sharing	—	—	—
NSGA-II	depth	—	crowding dist.	depth	—	crowding dist.
SPEA2	strength	—	kNN	strength	—	kNN
IBEA	—	binary indicator	—	—	binary ind.	—
HypE	—	I_H^h	—	depth	I_H^h	—
SMS-EMOA	—	—	—	depth-rank	I_H^1	—
DTLZ 2-obj	—	—	crowding	depth-rank	I_ϵ	sharing
DTLZ 3-obj	depth-rank	I_ϵ	kNN	rank	I_H^1	sharing
DTLZ 5-obj	rank	I_H^1	crowding	depth	I_H^1	—
WFG 2-obj	rank	—	crowding	depth-rank	I_H^1	—
WFG 3-obj	count	I_H^1	crowding	strength	I_H^1	sharing
WFG 5-obj	count	I_H^h	crowding	—	I_H^1	—

Experimental results

DTLZ			WFG		
2-obj $\Delta R = 126$	3-obj $\Delta R = 127$	5-obj $\Delta R = 107$	2-obj $\Delta R = 169$	3-obj $\Delta R = 130$	5-obj $\Delta R = 97$
$\texttt{Auto}_{\texttt{D2}}$ **(1339)**	$\texttt{Auto}_{\texttt{D3}}$ **(1500)**	$\texttt{Auto}_{\texttt{D5}}$ **(1002)**	$\texttt{Auto}_{\texttt{W2}}$ **(1692)**	$\texttt{Auto}_{\texttt{W3}}$ **(1375)**	$\texttt{Auto}_{\texttt{W5}}$ **(1170)**
SPEA2_{D2} (1562)	IBEA_{D3} (1719)	SMS_{D5} (1550)	SPEA2_{W2} (2097)	SMS_{W3} (1796)	SMS_{W5} (1567)
IBEA_{D2} (1940)	SMS_{D3} (1918)	IBEA_{D5} (1867)	$\text{NSGA-II}_{\text{W2}}$ (2542)	IBEA_{W3} (1843)	IBEA_{W5} (1746)
$\text{NSGA-II}_{\text{D2}}$ (2143)	HypE_{D3} (2019)	SPEA2_{D5} (2345)	SMS_{W2} (2621)	SPEA2_{W3} (2600)	SPEA2_{W5} (2747)
HypE_{D2} (2338)	SPEA2_{D3} (2164)	$\text{NSGA-II}_{\text{D5}}$ (2346)	IBEA_{W2} (2777)	$\text{NSGA-II}_{\text{W3}}$ (3315)	$\text{NSGA-II}_{\text{W5}}$ (3029)
SMS_{D2} (2406)	$\text{NSGA-II}_{\text{D3}}$ (2528)	HypE_{D5} (2674)	HypE_{W2} (2851)	HypE_{W3} (3431)	MOGA_{W5} (4268)
MOGA_{D2} (2970)	MOGA_{D3} (2851)	MOGA_{D5} (2915)	MOGA_{W2} (4320)	MOGA_{W3} (4540)	HypE_{W5} (4373)

AutoMOEA: Automatic Design

Automatic configuration (**irace**)
$+$ Flexible algorithmic framework (AutoMOEA)
$=$ Automatic design of state-of-the-art MOEAs

AutoMOEA: Automatic Design

Automatic configuration (**irace**)
+ Flexible algorithmic framework (AutoMOEA)
= Automatic design of state-of-the-art MOEAs

- Fair to compare with untuned traditional MOEAs?
- Why is our setup representative?

$\Rightarrow$ Different AutoMOEAs for termination criterion in FEs or seconds

- How do you define “state-of-the-art”?
- What is a “novel” MOEA?

AutoMOEA: Automatic Design

Automatic configuration (**irace**)
$+$ Flexible algorithmic framework (AutoMOEA)
$=$ Automatic design of state-of-the-art MOEAs

- Fair to compare with untuned traditional MOEAs?
- Why is our setup representative?

$\Rightarrow$ Different AutoMOEAs for termination criterion in FEs or seconds

- How do you define "state-of-the-art"?
- What is a "novel" MOEA?

Exactly!

Indicators may be conflicting

[Bezerra et al., 2020a]

Scenario (5, 40k)

I_H^{rd}	**Auto+** (0)	**SMS** (1)	IBEA (50)	MOEA/D (95)	SPEA2 (122)	CMA (125)
$I_{\epsilon+}$	**SMS** (0)	**IBEA** (5)	**Auto+** (21)	CMA (89)	MOEA/D (94)	SPEA2 (156)
I_{IGD}	**IBEA** (0)	**MOEA/D** (19)	**SMS** (25)	Auto+ (53)	SPEA2 (84)	CMA (105)

Indicators may be conflicting

[Bezerra et al., 2020a]

Scenario (5, 40*k*)

I_H^{rd}	**Auto+** (0)	**SMS** (1)	IBEA (50)	MOEA/D (95)	SPEA2 (122)	CMA (125)
$I_{\epsilon+}$	**SMS** (0)	**IBEA** (5)	**Auto+** (21)	CMA (89)	MOEA/D (94)	SPEA2 (156)
I_{IGD}	**IBEA** (0)	**MOEA/D** (19)	**SMS** (25)	Auto+ (53)	SPEA2 (84)	CMA (105)

Scenario (5, 40*k*)

I_H^{rd}	**Auto+** (0)	**SMS** (1)	**Auto-≺** (31)	IBEA (58)	MOEA/D (103)	SPEA2 (138)
$I_{\epsilon+}$	**Auto-≺** (0)	**SMS** (39)	IBEA (44)	Auto+ (61)	CMA (129)	MOEA/D (134)
I_{IGD}	**Auto-≺** (0)	IBEA (89)	MOEA/D (106)	SMS (113)	Auto+ (142)	SPEA2 (173)

Scenario (5, 40k)

I_H^{rd}	**Auto+** (0)	**SMS** (1)	IBEA (50)	MOEA/D (95)	SPEA2 (122)	CMA (125)
$I_{\epsilon+}$	**SMS** (0)	**IBEA** (5)	**Auto+** (21)	CMA (89)	MOEA/D (94)	SPEA2 (156)
I_{IGD}	**IBEA** (0)	**MOEA/D** (19)	**SMS** (25)	Auto+ (53)	SPEA2 (84)	CMA (105)

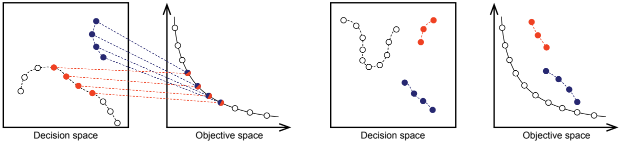
Scenario (5, 40k)

I_H^{rd}	**Auto+** (0)	**SMS** (1)	**Auto-ε** (31)	IBEA (58)	MOEA/D (103)	SPEA2 (138)
$I_{\epsilon+}$	**Auto-ε** (0)	**SMS** (39)	IBEA (44)	Auto+ (61)	CMA (129)	MOEA/D (134)
I_{IGD}	**Auto-ε** (0)	IBEA (89)	MOEA/D (106)	SMS (113)	Auto+ (142)	SPEA2 (173)

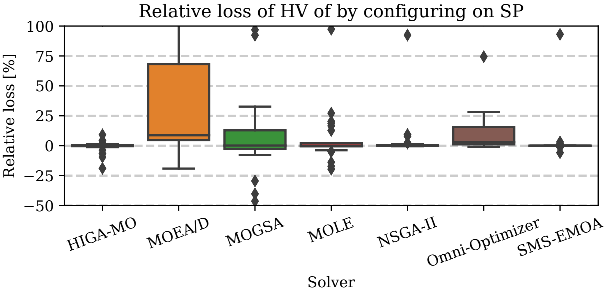
Scenario (10, 40k)

I_H^{rd}	**Auto-MO** (0)	**IBEA** (48)	SMS (104)	SPEA2 (114)	CMA (143)	Auto+ (143)
$I_{\epsilon+}$	**Auto-MO** (0)	**MOEA/D** (40)	IBEA (55)	Auto+ (98)	NSGA-III (149)	SMS (163)
I_{IGD}	**Auto-MO** (0)	IBEA (67)	NSGA-III (103)	SPEA2 (115)	NSGA-II (185)	HypE (201)

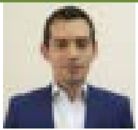
Use Case: Multi-modal multi-objective optimization [Rook et al.,'22]



- MMMOPS have Multiple global and local optima.
- Goal: Obtain *diversity* in decision space and *convergence* towards Pareto front.
- AAC for diversity (SP) results in a loss on convergence (HV) and *vice versa*.

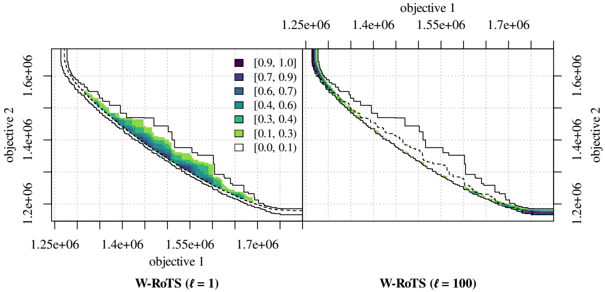


Interactive Automatic Design



Juan Esteban Diaz and Manuel López-Ibáñez,
Incorporating decision-maker's preferences into the automatic configuration of bi-objective optimisation algorithms,
European Journal of Operational Research, 289:3, 2021.
https://doi.org/10.1016/j.ejor.2020.07.059
☆ EJOR Editors' Choice Article, January 2021

Use the weighted hypervolume to guide
the automatic algorithm configuration of a bi-objective optimizer



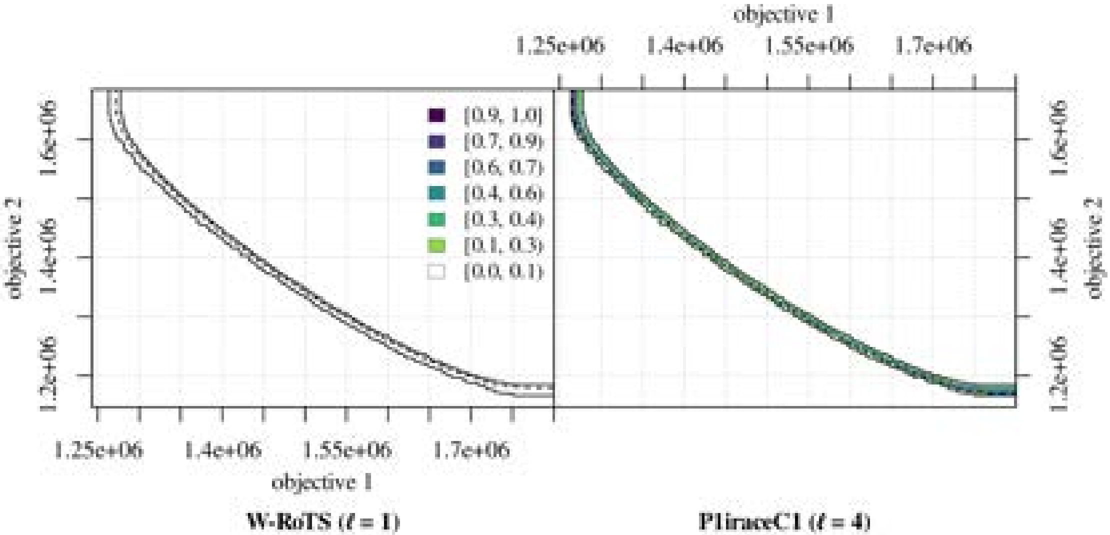
(1) The DM chooses one side, e.g., $\ell = 1$

(2) Compute regions $\mathcal{R}$ in favour

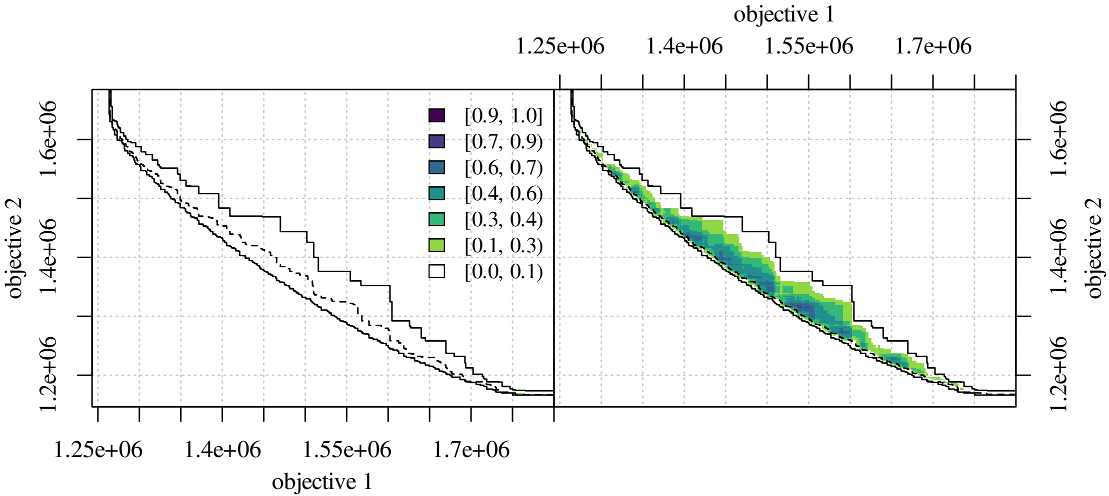
(2) Create weighted hypervolume (WHV) based on positive EAF differences

(3) Tune $\ell \in [1, 200]$ using irace guided by WHV (budget $= 500$ runs of W-RoTS)

Interactively Tuning the value of ℓ in W-RoTS [Diaz & López-Ibáñez, 2021]



Interactively Tuning the value of ℓ in W-RoTS [Diaz & López-Ibáñez, 2021]



W-RoTS (ℓ = 100) **P1iraceC2 (ℓ = 24)**

irace is clearly focusing on different regions

Part III

AAC for Improving Anytime Behaviour

Automatically Improving the Anytime Behavior of Optimization Algorithms with irace

Manuel López-Ibáñez and Thomas Stützle.
Automatically improving the anytime behaviour of optimisation algorithms.
European Journal of Operational Research, 2014. doi: 10.1016/j.ejor.2013.10.043.

Anytime Algorithm [Dean & Boddy, 1988]

- May be interrupted at any moment and returns a solution
- Keeps improving its solution until interrupted
- Eventually finds the optimal solution

Anytime Algorithm [Dean & Boddy, 1988]

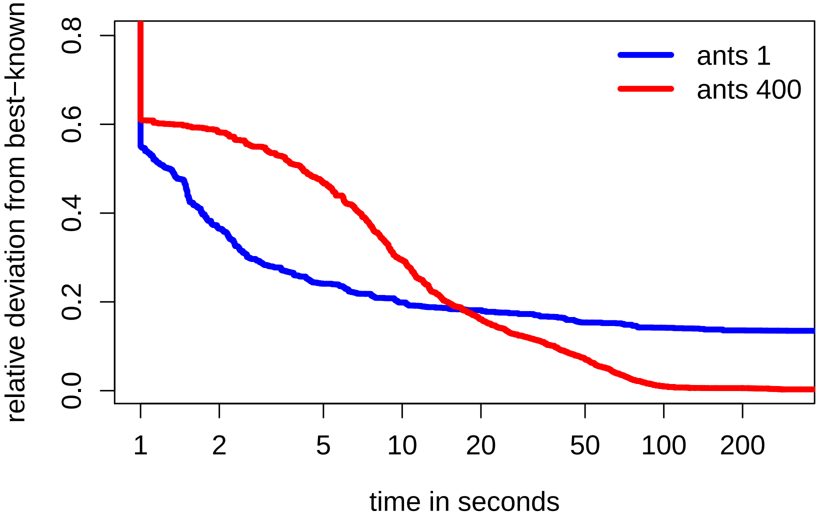
- May be interrupted at any moment and returns a solution
- Keeps improving its solution until interrupted
- Eventually finds the optimal solution

Good Anytime Behavior [Zilberstein, 1996]

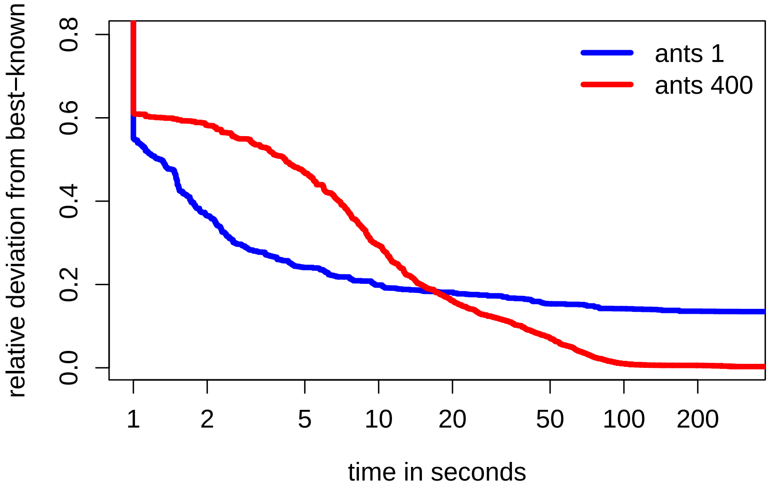
Algorithms with good *"anytime" behavior* produce as high quality result as possible at any moment of their execution.

Max-Min Ant System w/o LS

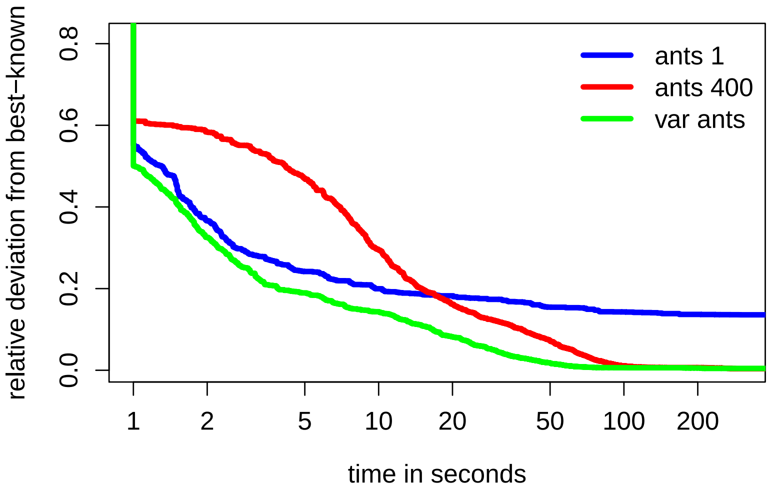
Solution-quality vs. time (SQT) curve / Performance profile



Algorithms with good *"anytime" behaviour* produce as high quality result as possible at any moment of their execution [Zilberstein, 1996]



Algorithms with good *"anytime" behaviour* produce as high quality result as possible at any moment of their execution [Zilberstein, 1996]



Improving Anytime Behaviour

How to improve the anytime behaviour of MMAS?

☞ Online parameter variation:

- Start with 1 ant, add 1 ant every iteration until 400 ants
- Start with $\beta = 10$, switch to $\beta = 2$ after 100 iterations
- ...

Improving Anytime Behaviour

How to improve the anytime behaviour of MMAS?

☞ Online parameter variation:

- Start with 1 ant, add 1 ant every iteration until 400 ants
- Start with $\beta = 10$, switch to $\beta = 2$ after 100 iterations
- ...

✗ More parameters!

✗ How to compare SQT curves? (Average solution quality plotted over time)

Classical (Human-intensive) Approach

1. Devise *many* online strategies for parameter variation
2. Run lots of experiments
3. Visually compare SQT plots

Classical (Human-intensive) Approach

1. Devise *many* online strategies for parameter variation
2. Run lots of experiments
3. Visually compare SQT plots

After one year and a master thesis: [Maur et al., 2010]

✔ Strategies for varying *ants*, β, or q_0 that significantly improve the anytime behaviour of MMAS on the TSP.

Classical (Human-intensive) Approach

1. Devise *many* online strategies for parameter variation
2. Run lots of experiments
3. Visually compare SQT plots

After one year and a master thesis: [Maur et al., 2010]

- ✔ Strategies for varying *ants*, β, or q_0 that significantly improve the anytime behaviour of MMAS on the TSP.
- ✘ Extremely time consuming
- ✘ Subjective / Bias

Automatically Improving the Anytime Behavior

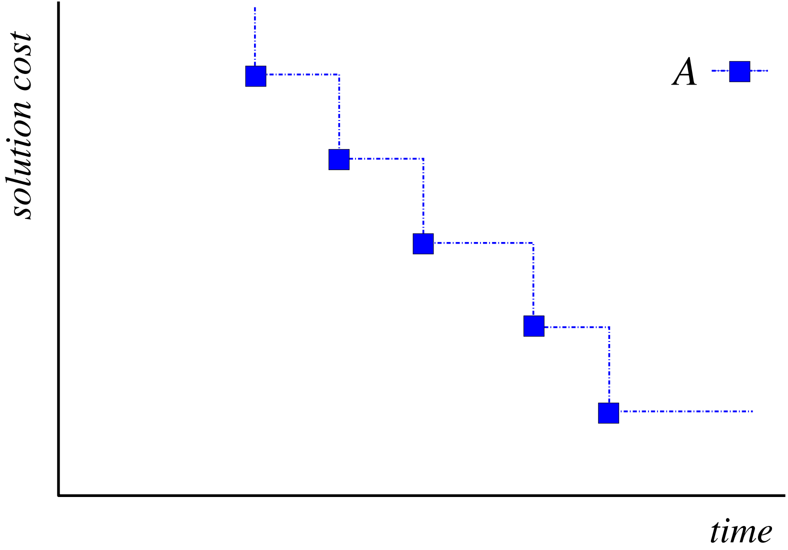
Online parameter control

- ✘ Which parameters to adapt? How? $\Rightarrow$ More parameters!
- ✔ Use irace (offline) to select the best parameter control strategies

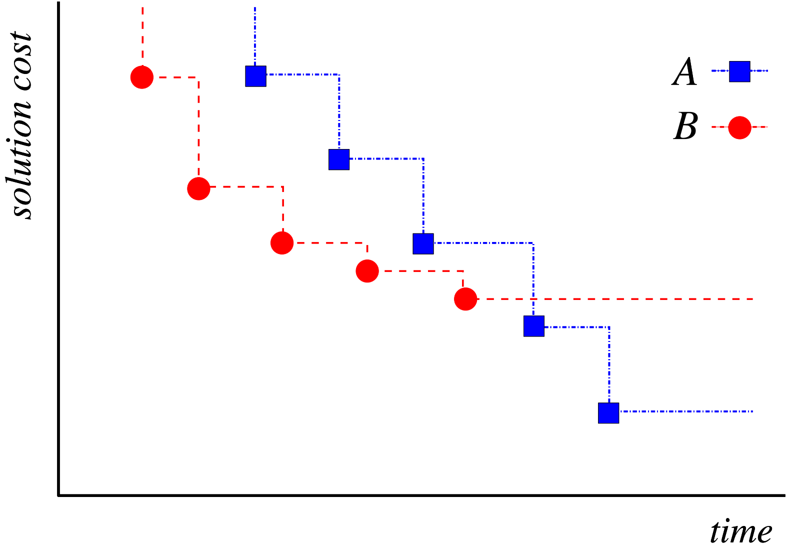
Improve Anytime Behavior

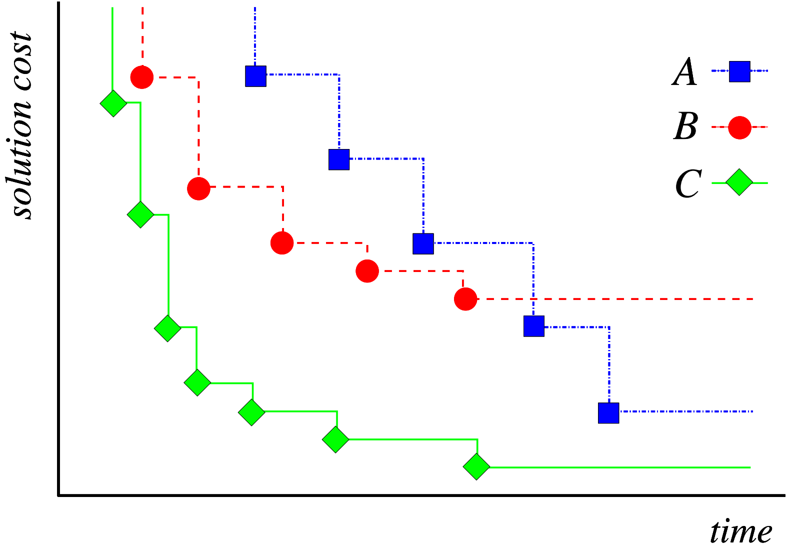
- ✔ More robust to different termination criteria
- ✘ How can irace compare SQT curves?

Automatically Improving the Anytime Behavior

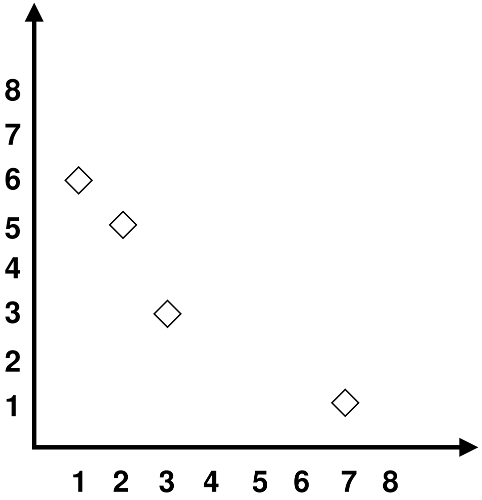


Automatically Improving the Anytime Behavior



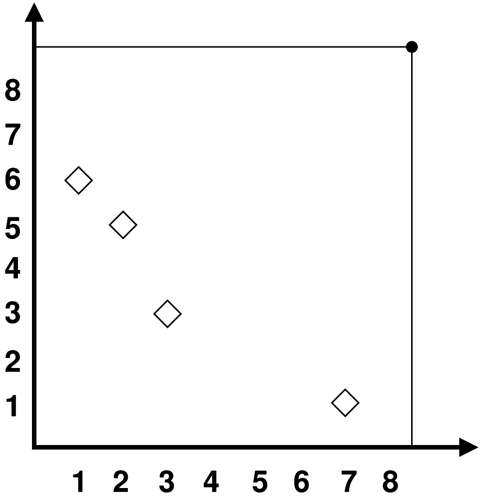

solution cost
A
B
C
time

Automatically Improving the Anytime Behavior

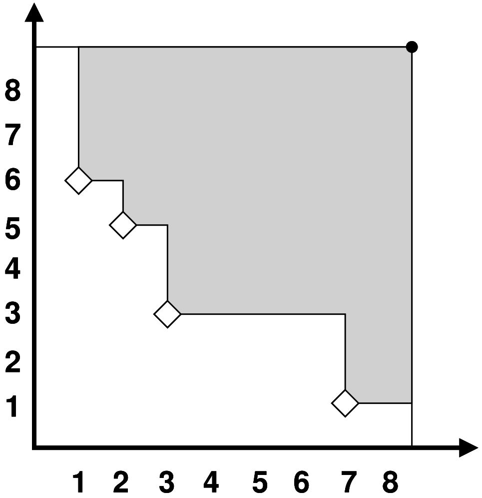


Hypervolume measure $\approx$ Anytime behaviour

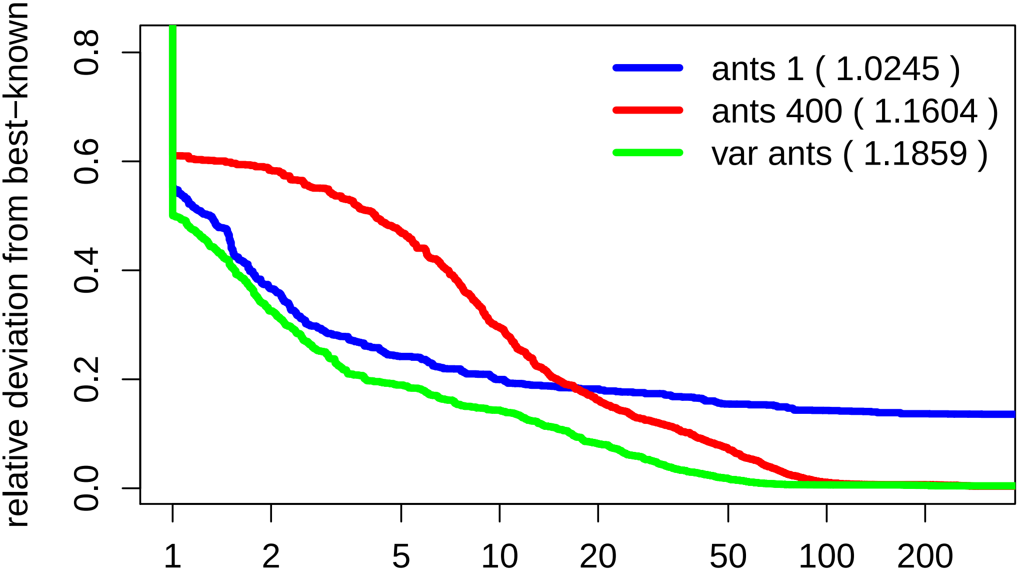
Automatically Improving the Anytime Behavior



Hypervolume measure ≈ Anytime behaviour



Hypervolume measure ≈ Anytime behaviour



relative deviation from best−known
0.0
0.2
0.4
0.6
0.8
1
2
5
10
20
50
100
200
ants 1 (1.0245)
ants 400 (1.1604)
var ants (1.1859)

Automatically Improving the Anytime Behavior

`irace` + hypervolume = automatically improving the anytime behavior of optimization algorithms

1. Run configuration until large stopping time
2. Compute hypervolume of SQT curve
3. Evaluate anytime behavior according to hypervolume

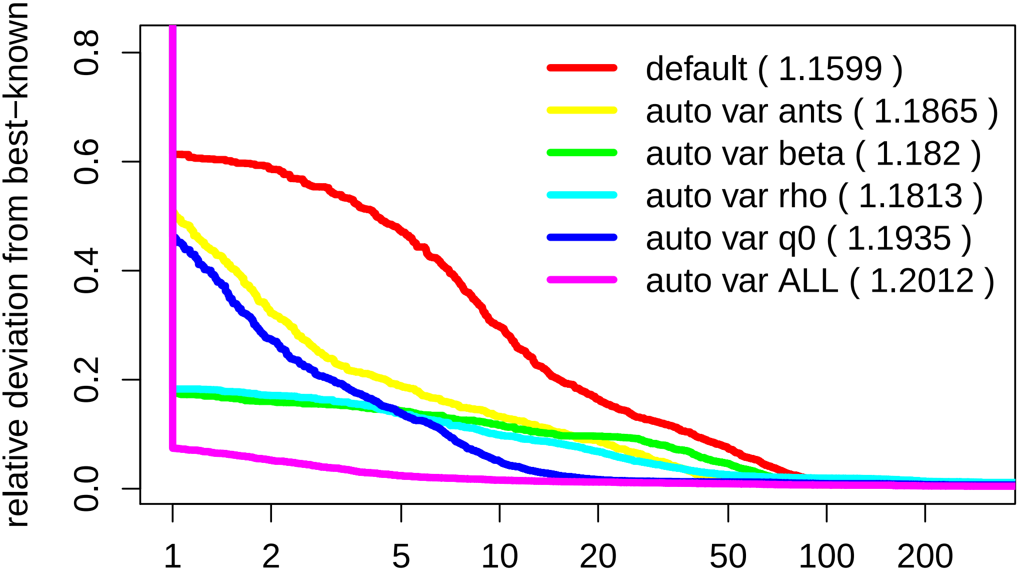
Automatically Improving the Anytime Behavior

`irace` + hypervolume = automatically improving the anytime behavior of optimization algorithms

1. Run configuration until large stopping time
2. Compute hypervolume of SQT curve
3. Evaluate anytime behavior according to hypervolume

- Hypervolume (multi-objective) optimization
  - ✔ Objectively defined comparison
  - ✔ Well-known performance measure
- Automatic configuration using irace
  - ✔ Most effort done by the computer
  - ✔ Best configurations selected by the computer: *Unbiased*

Scenario #1: Experimental comparison



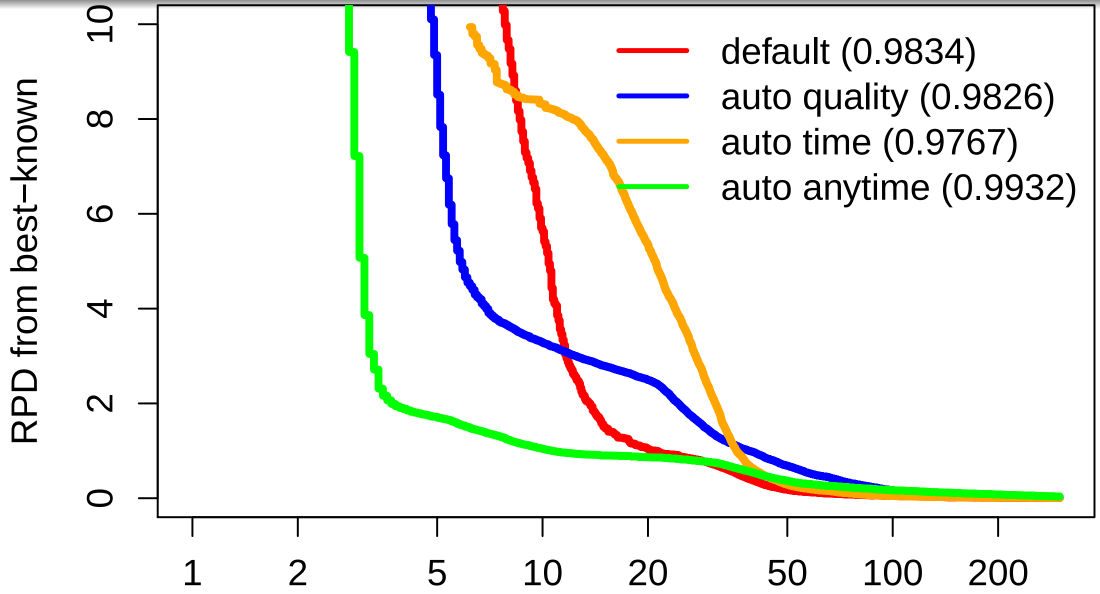
Scenario #2: SCIP

[López-Ibáñez & Stützle, 2014]

SCIP: an open-source mixed integer programming (MIP) solver [Achterberg, 2009]

- 200 parameters controlling search, heuristics, thresholds, ...
- Benchmark set: Winner determination problem for combinatorial auctions [Leyton-Brown et al., 2000]
  1 000 training + 1 000 testing instances
- Single run timeout: 300 seconds
- irace budget (*maxExperiments*): 5 000 runs

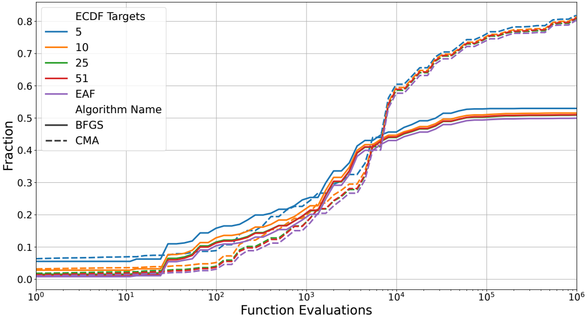
Scenario #2: SCIP



Hypervolume as a measure of anytime?

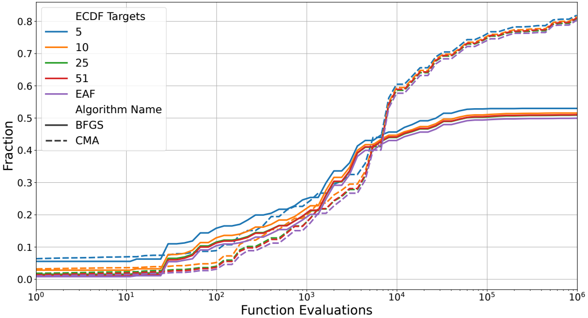
- What about the area under the target-based ECDFs?

COCO [Hansen et al., 2020]



Hypervolume as a measure of anytime?

- What about the area under the target-based ECDFs? COCO [Hansen et al., 2020]



☞ It is the same*!

* When the number of targets grows to infinity and except for a multiplication factor.

[López-Ibáñez, Vermetten, Dréo & Doerr, 2025]

- What if the user gives the stopping criterion before running?
  ☞ Then, tuning for various runtimes may be better [Branke & Elomari, 2011]

Part IV

AAC for Multiple Performance Objectives

In this part you will learn . . .

- in which situations you can (and should) use MO-AAC,
- which approaches you can use for MO-AAC, and
- how to properly deploy MO-AAC in your experiments.

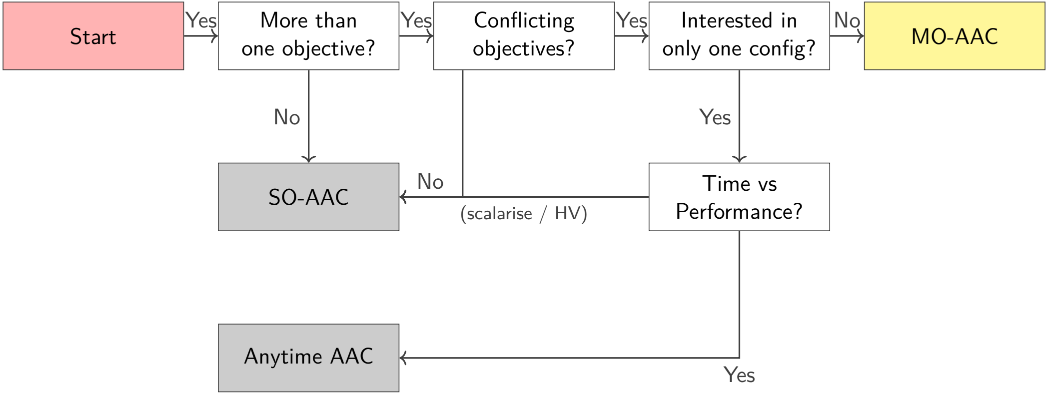
When to use MO-AAC?

- Prevent premature commitments towards preferences.
- Analyse trade-offs between objectives.

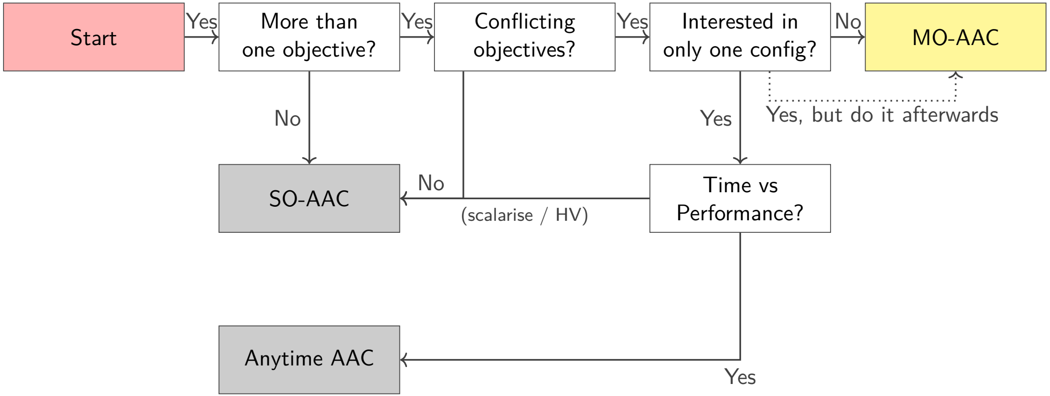
Objectives to consider

- Performance measures: > 100
- Robustness / stability: Variance
- Resources: Wall time, CPU time, Memory usage

When to use MO-AAC?



When to use MO-AAC?



Multi-Objective AAC

*Find a **set** of configurations for an algorithm that approaches the trade-off of the overall performance*

Formulated as multi-objective optimisation problem:

$$\Theta^* = \{\theta \in \Theta \mid \nexists \theta' \in \Theta \setminus \{\theta\} : \; p(A_{\theta'}, \mathcal{I}) \prec p(A_\theta, \mathcal{I})\}$$

Which methods to use?

Desired properties:

- Take multi-objective relations and challenges into account.
- Exploit evaluation on multiple problem instances.

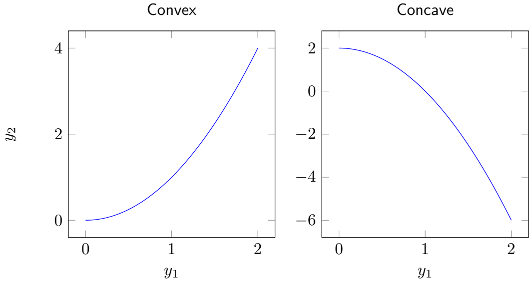
Existing methods:

- Off-the-shelf EMOAs.
- ParEGO: scalarisation with varying weights in combination with (SO)-AAC methods. [Knowles, 2006]
- Specialised MO-AAC frameworks:
  - MO-ParamILS,
  - MO-SMAC,
  - S-, I/S-, SPRINT-race.

MO-AAC Methods – Off-the-shelf EMOAs

- ✔ Many algorithms to choose from.
- ✘ Usually no good support for complex parameter spaces (mixed-type and dependencies/constraints).
- ✘ No mechanisms to efficiently handle the evaluation budget when configuring for multiple problem instances.

- Scalarize the objectives when interacting with a surrogate model but treat actual evaluation in the MO context.
- Vary the scalarisation weights each repetition.
- ✘ Does not work well for concave trade-offs.



MO-AAC Methods – MO-ParamILS [Blot et al., 2016]

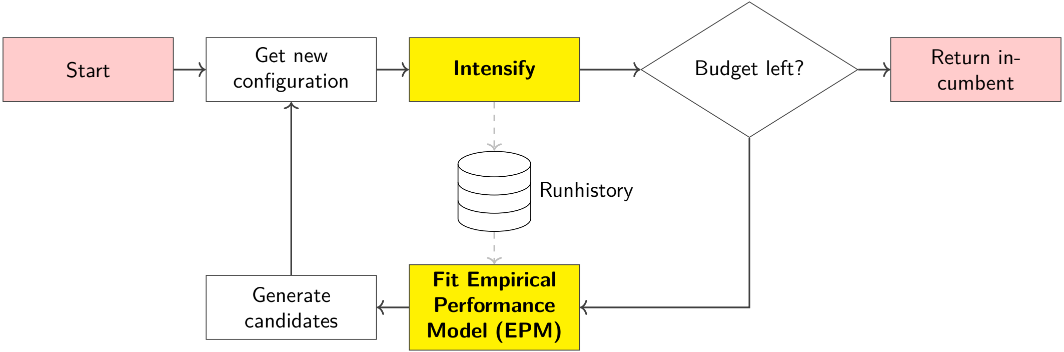
- Extension of ParamILS; Iterated Local Search (ILS) and keeps non-dominated configurations in an archive. [Hutter et al., 2007, 2009]

✔ Has an MO intensification[1] mechanism that works with ILS.

✘ Requires a discrete parameter space.

[1] Increasingly evaluate configurations on instances and stop them early to prevent wasteful computations.

- Pure MO extension of SMAC3 [Lindauer et al., 2022] (SMAC3 also supports ParEGO).
- Has a surrogate model for each objective and searches for configurations that complement the existing solution set to most by their improvement on the Hypervolume.
- ✔ Has an MO intensification mechanism
- ✔ Can handle complex parameter spaces and also does intensification.



MO-AAC Methods – S-, SPRINT-, I/S race [Zhang et al., 2013; Miranda et al., 2015; Zhang et al., 2016]

- Replaces the F-test in F-race with a new statistical test and races to obtain a ND set of configurations.
- ✘ Requires substantial evaluation budgets.

MO-AAC Methods

- Which ones is best? $\rightarrow$Just as with EMOA, there is not one that rules them all.
- Also depends on configuration budgets and used programming language.
- We like MO-SMAC though ;)

Best-practices for MO-AAC

- We got a problem scenario

Best-practices for MO-AAC

- We got a problem scenario
- We got an MO-AAC method

Best-practices for MO-AAC

- We got a problem scenario
- We got an MO-AAC method
- What elso to consider? . . .

Best-practices for MO-AAC

- We got a problem scenario
- We got an MO-AAC method
- What elso to consider? . . . The setup!

Most of the regular AAC best-practices still apply

- Perform multiple runs of the configurator due to their stochastic behaviour.
- Carefully think the parameter configuration space through
  - What are good bounds for parameters? Open problem!
- Run on the same hardware and software + check filesystem throughput
- Use (various) train-test splits; configure on training set; select best from all runs based on training set; report performance on test set.
- Report configurations found when configuring on whole instance set.

Best-practices especially for MO-AAC

- How to quantify the performance of the outcome?
  - Performance indicators, like Hypervolume.
  - Decision maker's statisfaction.
- Configuration sets found over multiple runs can be complementary to each other!
- Select those that are non-dominating on their performance on the training set.

Aggregating results from multiple independent runs

SO-AAC

Run $1 \longrightarrow \theta_1$

$\vdots$

Run $n \longrightarrow \theta_n$

θ^*

MO-AAC

Run $1 \longrightarrow \{\theta_{1,1}, \ldots, \theta_{1,4}\}$

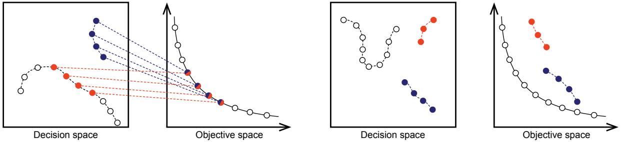
$\vdots$

Run $n \longrightarrow \{\theta_{n,1}, \ldots, \theta_{n,8}\}$

$\{\theta_{1,1}, \ldots, \theta_{n,5}\}$

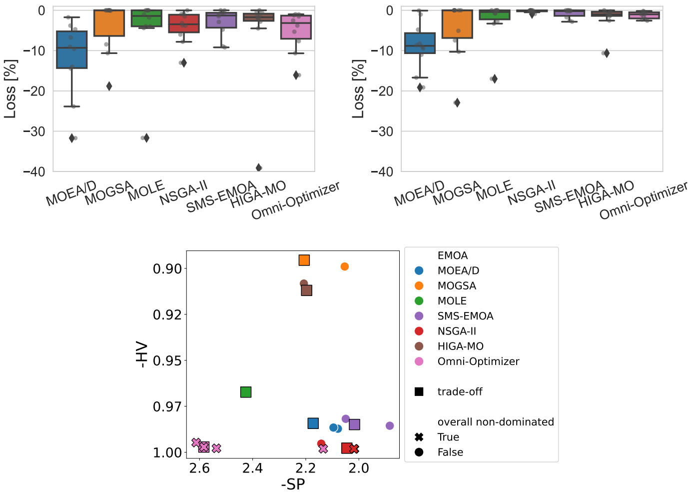
Use-cases

Use-case 1: Multi-Modal MOPs [Preuß et al., 2024]



- Recall the loss in diversity (SP) and convergence (HV) from [Rook et al., 2022]
- Can we mitigate the trade-off between SP and HV with MO-AAC?

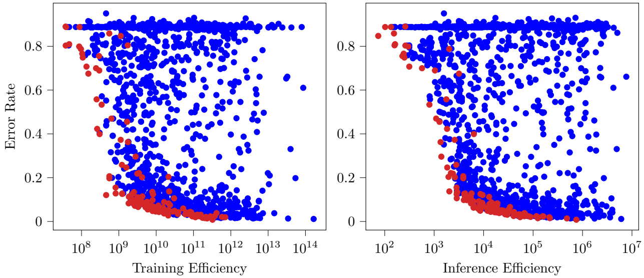
How does the trade-off between SP and HV look like?



- Omni-Optimizer achieved the best overall performance

Use-case 2: Sparse Neural Networks

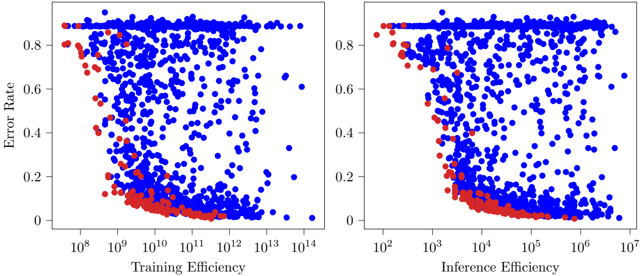
- NNs with sparsely connected layers; optimised during training. [Mocanu et al., 2018]



Use-case 2: Sparse Neural Networks

- NNs with sparsely connected layers; optimised during training. [Mocanu et al., 2018]

- Experiment: Observe performance of various sparsity levels with same model.
- Original claim: Sparse NNs are more efficient and perform same as dense NNs.

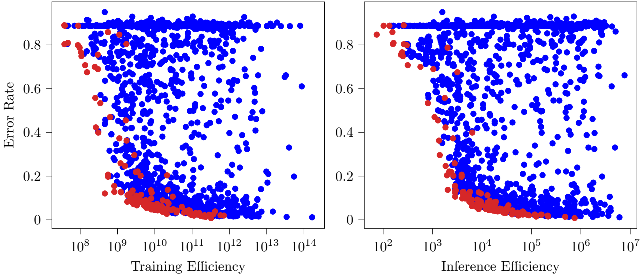


Use-case 2: Sparse Neural Networks

- NNs with sparsely connected layers; optimised during training. [Mocanu et al., 2018]

- Experiment: Observe performance of various sparsity levels with same model.
- Original claim: Sparse NNs are more efficient and perform same as dense NNs.

- MO-AAC experiment: Configure model for performance and efficiency.
- Observation: Small dense networks perform equally and are much more efficient.



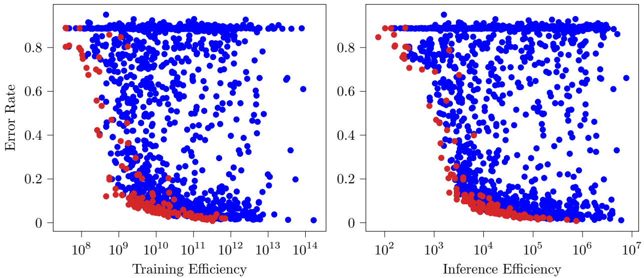
Use-case 2: Sparse Neural Networks

- NNs with sparsely connected layers; optimised during training. [Mocanu et al., 2018]

- Experiment: Observe performance of various sparsity levels with same model.
- Original claim: Sparse NNs are more efficient and perform same as dense NNs.

- MO-AAC experiment: Configure model for performance and efficiency.
- Observation: Small dense networks perform equally and are much more efficient.

Early commitment does not show the actual trade-off.



In this part you learned . . .

- in which situations you can (and should) use MO-AAC,
- which approaches you can use for MO-AAC, and
- how to properly deploy MO-AAC in your experiments.

Part V

Wrap-up

Recap

Part II: AAC for Multi-Objective Optimization Algorithms

Part III: AAC for Improving Anytime Behaviour

Part IV: AAC for Multiple Performance Objectives

Considerations

- AAC is systematic, but not exhaustive $\rightarrow$ no guarantee of optimal solution.
- AAC gives equal opportunity to algorithms to behave at their best for a given problem.
- What about ensembles of different configurators (or configurations of configurators)?

Take aways

- Do not use default parameters. Always configure.
- AAC does not only optimize, it is also an analysis tool.
- Who is configuring the configurator?

Links to configurator frameworks:

irace https://mlopez-ibanez.github.io/irace/

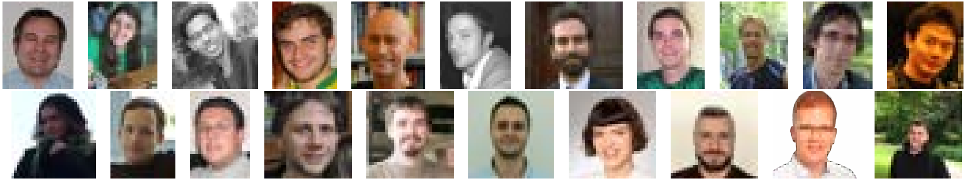
MO-SMAC https://github.com/jeroenrook/SMAC3/tree/mosmac-anon[2]

[2]Soon to be merged with SMAC3.

Acknowledgments

This tutorial has benefited from collaborations and discussions with our colleagues:

Thomas Stützle, Leslie Pérez Cáceres, Prasanna Balaprakash, Leonardo Bezerra, Mauro Birattari, Jérémie Dubois-Lacoste, Alberto Franzin, Holger H. Hoos, Frank Hutter, Kevin Leyton-Brown, Tianjun Liao, Marie-Eléonore Marmion, Franco Mascia, Marco Montes de Oca, Federico Pagnozzi, Zhi Yuan, Marcus Ritt, Marcelo De Souza, Carolin Benjamins, Jakob Bossek, Marius Lindauer, Oliver Preuß



References I

T. Achterberg. SCIP: Solving constraint integer programs. *Mathematical Programming Computation*, 1(1):1–41, July 2009.

C. Audet, J. Bigeon, D. Cartier, S. Le Digabel, and L. Salomon. Performance indicators in multiobjective optimization. *European Journal of Operational Research*, 292 (2):397–422, July 2021. ISSN 0377-2217. doi: 10.1016/j.ejor.2020.11.016.

J. Bader and E. Zitzler. HypE: An algorithm for fast hypervolume-based many-objective optimization. *Evolutionary Computation*, 19(1):45–76, 2011. doi: 10.1162/EVCO_a_00009.

N. Beume, B. Naujoks, and M. T. M. Emmerich. SMS-EMOA: Multiobjective selection based on dominated hypervolume. *European Journal of Operational Research*, 181(3):1653–1669, 2007. doi: 10.1016/j.ejor.2006.08.008.

L. C. T. Bezerra, M. López-Ibáñez, and T. Stützle. Automatic component-wise design of multi-objective evolutionary algorithms. *IEEE Transactions on Evolutionary Computation*, 20(3):403–417, 2016. doi: 10.1109/TEVC.2015.2474158.

L. C. T. Bezerra, M. López-Ibáñez, and T. Stützle. Automatically designing state-of-the-art multi- and many-objective evolutionary algorithms. *Evolutionary Computation*, 28(2):195–226, 2020a. doi: 10.1162/evco_a_00263.

L. C. T. Bezerra, M. López-Ibáñez, and T. Stützle. Automatic configuration of multi-objective optimizers and multi-objective configuration. In T. Bartz-Beielstein, B. Filipič, P. Korošec, and E.-G. Talbi, editors, *High-Performance Simulation-Based Optimization*, pages 69–92. Springer International Publishing, Cham, Switzerland, 2020b. doi: 10.1007/978-3-030-18764-4_4.

B. Bischl, M. Lang, J. Bossek, L. Judt, J. Richter, T. Kuehn, and E. Studerus. *mlr: Machine Learning in R*, 2013. URL http://cran.r-project.org/package=mlr. R package.

B. Bischl, M. Lang, L. Kotthoff, J. Schiffner, J. Richter, E. Studerus, G. Casalicchio, and Z. M. Jones. mlr: Machine learning in R. *Journal of Machine Learning Research*, 17(170):1–5, 2016.

A. Blot, H. H. Hoos, L. Jourdan, M.-E. Kessaci-Marmion, and H. Trautmann. MO-ParamILS: A multi-objective automatic algorithm configuration framework. In P. Festa, M. Sellmann, and J. Vanschoren, editors, *Learning and Intelligent Optimization, 10th International Conference, LION 10*, volume 10079 of *Lecture Notes in Computer Science*, pages 32–47. Springer, Cham, Switzerland, 2016. doi: 10.1007/978-3-319-50349-3_3.

J. Branke and J. Elomari. Simultaneous tuning of metaheuristic parameters for various computing budgets. In Krasnogor & Lanzi [2011], pages 263–264. doi: 10.1145/2001858.2002006.

N. Dang and C. Doerr. Hyper-parameter tuning for the $(1 + (\lambda, \lambda))$ GA. In López-Ibáñez et al. [2019], pages 889–897. ISBN 978-1-4503-6111-8. doi: 10.1145/3321707.3321725.

T. Dean and M. S. Boddy. An analysis of time-dependent planning. In H. E. Shrobe, T. M. Mitchell, and R. G. Smith, editors, *Proceedings of the 7th National Conference on Artificial Intelligence, AAAI-88*, pages 49–54. AAAI Press/MIT Press, Menlo Park, CA, 1988. URL http://www.aaai.org/Conferences/AAAI/aaai88.php.

K. Deb and H. Jain. An evolutionary many-objective optimization algorithm using reference-point-based nondominated sorting approach, part I: Solving problems with box constraints. *IEEE Transactions on Evolutionary Computation*, 18(4):577–601, 2014.

K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan. A fast and elitist multi-objective genetic algorithm: NSGA-II. *IEEE Transactions on Evolutionary Computation*, 6(2): 182–197, 2002. doi: 10.1109/4235.996017.

J. E. Diaz and M. López-Ibáñez. Incorporating decision-maker's preferences into the automatic configuration of bi-objective optimisation algorithms. *European Journal of Operational Research*, 289(3):1209–1222, 2021. doi: 10.1016/j.ejor.2020.07.059.

C. M. Fonseca and P. J. Fleming. Genetic algorithms for multiobjective optimization: Formulation, discussion and generalization. In S. Forrest, editor, *Proceedings of the Fifth International Conference on Genetic Algorithms (ICGA'93)*, pages 416–423. Morgan Kaufmann Publishers, 1993. ISBN 1-55860-299-2.

G. Francesca, M. Brambilla, A. Brutschy, L. Garattoni, R. Miletitch, G. Podevijn, A. Reina, T. Soleymani, M. Salvaro, C. Pinciroli, F. Mascia, V. Trianni, and M. Birattari. AutoMoDe-Chocolate: Automatic design of control software for robot swarms. *Swarm Intelligence*, 2015. doi: 10.1007/s11721-015-0107-9.

T. Friedrich, F. Quinzan, and M. Wagner. Escaping large deceptive basins of attraction with heavy-tailed mutation operators. In H. E. Aguirre and K. Takadama, editors, *Proceedings of the Genetic and Evolutionary Computation Conference, GECCO 2018*, pages 293–300. ACM Press, New York, NY, 2018. doi: 10.1145/3205455.3205515.

G. T. Hall, P. S. Oliveto, and D. Sudholt. On the impact of the cutoff time on the performance of algorithm configurators. In López-Ibáñez et al. [2019], pages 907–915. ISBN 978-1-4503-6111-8. doi: 10.1145/3321707.3321879.

N. Hansen, A. Auger, R. Ros, O. Mersmann, T. Tušar, and D. Brockhoff. COCO: A platform for comparing continuous optimizers in a black-box setting. *Optimization Methods and Software*, 36(1):1–31, 2020. doi: 10.1080/10556788.2020.1808977.

F. Hutter and S. Ramage. Manual for smac version v2. 10.03-master. *Vancouver: Department of Computer Science University of British Columbia*, 2015. URL https://www.cs.ubc.ca/labs/algorithms/Projects/SMAC/v2.10.03/manual.pdf.

F. Hutter, H. H. Hoos, and T. Stützle. Automatic algorithm configuration based on local search. In R. C. Holte and A. Howe, editors, *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 1152–1157. AAAI Press/MIT Press, Menlo Park, CA, 2007.

F. Hutter, H. H. Hoos, K. Leyton-Brown, and T. Stützle. ParamILS: an automatic algorithm configuration framework. *Journal of Artificial Intelligence Research*, 36: 267–306, Oct. 2009. doi: 10.1613/jair.2861.

C. Igel, N. Hansen, and S. Roth. Covariance matrix adaptation for multi-objective optimization. *Evolutionary Computation*, 15(1):1–28, 2007.

J. D. Knowles. ParEGO: A hybrid algorithm with on-line landscape approximation for expensive multiobjective optimization problems. *IEEE Transactions on Evolutionary Computation*, 10(1):50–66, 2006. doi: 10.1109/TEVC.2005.851274.

J. D. Knowles and D. Corne. Approximating the nondominated front using the Pareto archived evolution strategy. *Evolutionary Computation*, 8(2):149–172, 2000. doi: 10.1162/106365600568167.

J. D. Knowles, L. Thiele, and E. Zitzler. A tutorial on the performance assessment of stochastic multiobjective optimizers. TIK-Report 214, Computer Engineering and Networks Laboratory (TIK), Swiss Federal Institute of Technology (ETH), Zürich, Switzerland, Feb. 2006. Revised version.

N. Krasnogor and P. L. Lanzi, editors. *Genetic and Evolutionary Computation Conference, GECCO 2011, Proceedings, Dublin, Ireland, July 12-16, 2011*. ACM Press, New York, NY, 2011.

S. Kukkonen and J. Lampinen. GDE3: the third evolution step of generalized differential evolution. In *Proceedings of the 2005 Congress on Evolutionary Computation (CEC 2005)*, pages 443–450, Piscataway, NJ, Sept. 2005. IEEE Press.

M. Lang, H. Kotthaus, P. Marwedel, C. Weihs, J. Rahnenführer, and B. Bischl. Automatic model selection for high-dimensional survival analysis. *Journal of Statistical Computation and Simulation*, 85(1):62–76, 2014. doi: 10.1080/00949655.2014.929131.

K. Leyton-Brown, M. Pearson, and Y. Shoham. Towards a universal test suite for combinatorial auction algorithms. In A. Jhingran et al., editors, *ACM Conference on Electronic Commerce (EC-00)*, pages 66–76. ACM Press, New York, NY, 2000. doi: 10.1145/352871.352879.

H. Li and Q. Zhang. Multiobjective optimization problems with complicated Pareto sets, MOEA/D and NSGA-II. *IEEE Transactions on Evolutionary Computation*, 13 (2):284–302, 2009.

M. T. Lindauer, K. Eggensperger, M. Feurer, A. Biedenkapp, D. Deng, C. Benjamins, T. Ruhkopf, R. Sass, and F. Hutter. SMAC3: A versatile bayesian optimization package for hyperparameter optimization. *Journal of Machine Learning Research*, 23:1–9, 2022.

M. López-Ibáñez and T. Stützle. The automatic design of multi-objective ant colony optimization algorithms. *IEEE Transactions on Evolutionary Computation*, 16(6): 861–875, 2012. doi: 10.1109/TEVC.2011.2182651.

M. López-Ibáñez and T. Stützle. Automatically improving the anytime behaviour of optimisation algorithms. *European Journal of Operational Research*, 235(3):569–582, 2014. doi: 10.1016/j.ejor.2013.10.043.

References IV

M. López-Ibáñez, A. Auger, and T. Stützle, editors. *Proceedings of the Genetic and Evolutionary Computation Conference, GECCO 2019, Prague, Czech Republic, July 13-17, 2019*. ACM Press, New York, NY, 2019. ISBN 978-1-4503-6111-8. doi: 10.1145/3321707.

M. López-Ibáñez, D. Vermetten, J. Dréo, and C. Doerr. Using the empirical attainment function for analyzing single-objective black-box optimization algorithms. *IEEE Transactions on Evolutionary Computation*, 2025. Accepted, pre-print available at https://doi.org/10.48550/arXiv.2404.02031.

R. Martín-Santamaría, M. López-Ibáñez, T. Stützle, and J. M. Colmenar. On the automatic generation of metaheuristic algorithms for combinatorial optimization problems. *European Journal of Operational Research*, 2024. doi: 10.1016/j.ejor.2024.06.001.

M. Maur, M. López-Ibáñez, and T. Stützle. Pre-scheduled and adaptive parameter variation in $\mathcal{MAX}$-$\mathcal{MIN}$ Ant System. In H. Ishibuchi et al., editors, *Proceedings of the 2010 Congress on Evolutionary Computation (CEC 2010)*, pages 3823–3830, Piscataway, NJ, 2010. IEEE Press. doi: 10.1109/CEC.2010.5586332.

P. Miranda, R. M. Silva, and R. B. Prudêncio. I/S-Race: An iterative multi-objective racing algorithm for the SVM parameter selection problem. In *European Symposium on Artificial Neural Networks, ESSAN*, pages 573–578, 2015.

D. C. Mocanu, E. Mocanu, P. Stone, P. H. Nguyen, M. Gibescu, and A. Liotta. Scalable training of artificial neural networks with adaptive sparse connectivity inspired by network science. *Nature Communications*, 9(1):2383, Dec. 2018. ISSN 2041-1723. doi: 10.1038/s41467-018-04316-3.

A. J. Nebro, M. López-Ibáñez, C. Barba-González, and J. García-Nieto. Automatic configuration of NSGA-II with jMetal and irace. In M. López-Ibáñez, A. Auger, and T. Stützle, editors, *Proceedings of the Genetic and Evolutionary Computation Conference, GECCO Companion 2019*, pages 1374–1381. ACM Press, New York, NY, 2019. ISBN 978-1-4503-6748-6. doi: 10.1145/3319619.3326832.

L. Pérez Cáceres, F. Pagnozzi, A. Franzin, and T. Stützle. Automatic configuration of GCC using irace: Supplementary material. http://iridia.ulb.ac.be/supp/IridiaSupp2017-009/, 2017.

O. L. Preuß, J. Rook, and H. Trautmann. On the Potential of Multi-objective Automated Algorithm Configuration on Multi-modal Multi-objective Optimisation Problems. In S. Smith, J. Correia, and C. Cintrano, editors, *Applications of Evolutionary Computation*, volume 14634, pages 305–321. Springer Nature Switzerland, Cham, 2024. ISBN 978-3-031-56851-0 978-3-031-56852-7. doi: 10.1007/978-3-031-56852-7_20.

T. Robič and B. Filipič. DEMO: Differential evolution for multiobjective optimization. In C. A. Coello Coello, A. Hernández Aguirre, and E. Zitzler, editors, *Evolutionary Multi-criterion Optimization, EMO 2005*, volume 3410 of *Lecture Notes in Computer Science*, pages 520–533. Springer, Heidelberg, Germany, 2005.

J. Rook, H. Trautmann, J. Bossek, and C. Grimme. On the potential of automated algorithm configuration on multi-modal multi-objective optimization problems. In *Proceedings of the Genetic and Evolutionary Computation Conference Companion*, pages 356–359, Boston Massachusetts, July 2022. ACM. ISBN 978-1-4503-9268-6. doi: 10.1145/3520304.3528998.

References V

J. Rook, C. Benjamins, J. Bossek, H. Trautmann, H. H. Hoos, and M. Lindauer. MO-SMAC: Multi-objective Sequential Model-based Algorithm Configuration. *Evolutionary Computation Journal [Submitted]*, 2024.

K. Tagawa, H. Shimizu, and H. Nakamura. Indicator-based differential evolution using exclusive hypervolume approximation and parallelization for multi-core processors. In Krasnogor & Lanzi [2011], pages 657–664.

T. Tušar and B. Filipič. Differential evolution versus genetic algorithms in multiobjective optimization. In S. Obayashi et al., editors, *Evolutionary Multi-criterion Optimization, EMO 2007*, volume 4403 of *Lecture Notes in Computer Science*, pages 257–271. Springer, Heidelberg, Germany, 2007.

T. Zhang, M. Georgiopoulos, and G. C. Anagnostopoulos. S-Race: A multi-objective racing algorithm. In C. Blum and E. Alba, editors, *Proceedings of the Genetic and Evolutionary Computation Conference, GECCO 2013*, pages 1565–1572. ACM Press, New York, NY, 2013. ISBN 978-1-4503-1963-8.

T. Zhang, M. Georgiopoulos, and G. C. Anagnostopoulos. Multi-objective model selection via racing. *IEEE Transactions on Cybernetics*, 46(8):1863–1876, 2016.

S. Zilberstein. Using anytime algorithms in intelligent systems. *AI Magazine*, 17(3):73–83, 1996. doi: 10.1609/aimag.v17i3.1232.

E. Zitzler and S. Künzli. Indicator-based selection in multiobjective search. In X. Yao et al., editors, *Parallel Problem Solving from Nature – PPSN VIII*, volume 3242 of *Lecture Notes in Computer Science*, pages 832–842. Springer, Heidelberg, Germany, 2004.

E. Zitzler, M. Laumanns, and L. Thiele. SPEA2: Improving the strength Pareto evolutionary algorithm for multiobjective optimization. In K. C. Giannakoglou, D. T. Tsahalis, J. Periaux, K. D. Papaliliou, and T. Fogarty, editors, *Evolutionary Methods for Design, Optimisation and Control*, pages 95–100. CIMNE, Barcelona, Spain, 2002. ISBN 84-89925-97-6.

E. Zitzler, L. Thiele, M. Laumanns, C. M. Fonseca, and V. Grunert da Fonseca. Performance assessment of multiobjective optimizers: an analysis and review. *IEEE Transactions on Evolutionary Computation*, 7(2):117–132, 2003. doi: 10.1109/TEVC.2003.810758.

E. Zitzler, L. Thiele, and J. Bader. On set-based multiobjective optimization. *IEEE Transactions on Evolutionary Computation*, 14(1):58–79, 2010. doi: 10.1109/TEVC.2009.2016569.

Back-up slides

- irace, MO-SMAC working

The irace Package

Manuel López-Ibáñez, Jérémie Dubois-Lacoste, Leslie Pérez Cáceres, Thomas Stützle, and Mauro Birattari.
The irace package: Iterated Racing for Automatic Algorithm Configuration.
Operations Research Perspectives, 3:43–58, 2016. doi: 10.1016/j.orp.2016.09.002
https://mlopez-ibanez.github.io/irace/

The irace Package

Manuel López-Ibáñez, Jérémie Dubois-Lacoste, Leslie Pérez Cáceres, Thomas Stützle, and Mauro Birattari.
The irace package: Iterated Racing for Automatic Algorithm Configuration.
Operations Research Perspectives, 3:43–58, 2016. doi: 10.1016/j.orp.2016.09.002
https://mlopez-ibanez.github.io/irace/

- Implementation of Iterated Racing in R

  Goal 1: Flexible

  Goal 2: Easy to use

The irace Package

Manuel López-Ibáñez, Jérémie Dubois-Lacoste, Leslie Pérez Cáceres, Thomas Stützle, and Mauro Birattari.
The irace package: Iterated Racing for Automatic Algorithm Configuration.
Operations Research Perspectives, 3:43–58, 2016. doi: 10.1016/j.orp.2016.09.002
https://mlopez-ibanez.github.io/irace/

- Implementation of Iterated Racing in R

  Goal 1: Flexible

  Goal 2: Easy to use

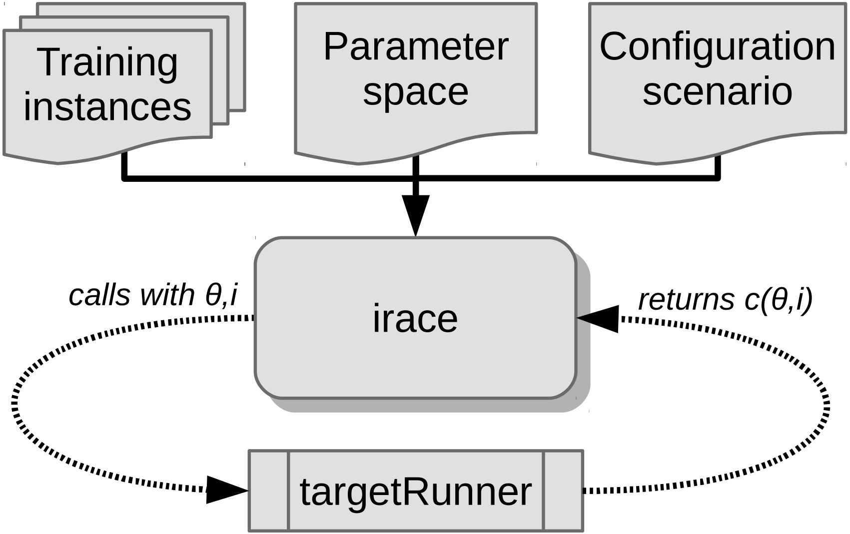
- R package available at CRAN (GNU/Linux, Windows, OSX)

  http://cran.r-project.org/package=irace

- Use it through the command-line: (see `irace --help`)

```
irace --max-experiments 1000 --param-file parameters.txt
```

✔ No knowledge of R needed


Training instances
Parameter space
Configuration scenario
irace
calls with θ,i
returns c(θ,i)
targetRunner

The irace Package: Instances

- TSP instances

```
$ dir Instances/
3000-01.tsp 3000-02.tsp 3000-03.tsp ...
```

- Continuous functions

```
$ cat instances.txt
function=1 dimension=100
function=2 dimension=100
...
```

- Parameters for an instance generator

```
$ cat instances.txt
I1 --size 100 --num-clusters 10 --sym yes --seed 1
I2 --size 100 --num-clusters 5 --sym no --seed 1
...
```

- Script / R function that generates instances

☞ if you need this, tell us!

The irace Package: Parameter space

- Categorical (`c`), ordinal (`o`), integer (`i`) and real (`r`)
- Subordinate parameters (`| condition`)
- Logarithmic scale (`,log`) *(irace 3.0)*

```
$ cat parameters.txt
```

```
# Name      Label/switch       Type   Domain            Condition
LS          "--localsearch "    c     (SA, TS, II)
rate        "--rate="           o     (low, med, high)
population  "--pop "          i,log   (1, 100)
temp        "--temp "           r     (0.5, 1)          | LS == "SA"
```

- For real parameters, number of decimal places is controlled by option *digits* (`--digits`)

The irace Package: Options

- *maxExperiments* (*maxTime*): maximum number of runs (or overall time) of the target algorithm (tuning budget)
- *testType*: either F-test or t-test

The irace Package: target-runner

- A script/program that calls the software to be tuned:

  `./target-runner` *configID instanceID seed instance configuration*

e.g. :

```
./target-runner 2 1 1234567 3000-01.tsp --localsearch SA ...
```

- An R function

Flexibility: If there is something you cannot tune, let us know!

The irace Package: Other features

1. Initial configurations (e.g., default configuration)
2. Parallel evaluation: multiple CPUs, MPI, batch job clusters (SGE, PBS, Torque, Slurm)
3. Forbidden configurations (+ rejection):

   ```
   popsize < 5 & LS == "SA"
   ```

4. Recovery file: allows resuming an interrupted irace run
5. Test instances
6. Repair configurations before being evaluated
7. Adaptive capping (for runtime minimization)

The irace Package

Last version 3.5 (23/10/2022)

A detailed user-guide / tutorial:

https://cran.r-project.org/web/packages/irace/vignettes/irace-package.pdf

GitHub: https://github.com/MLopez-Ibanez/irace

Google group

https://groups.google.com/d/forum/irace-package

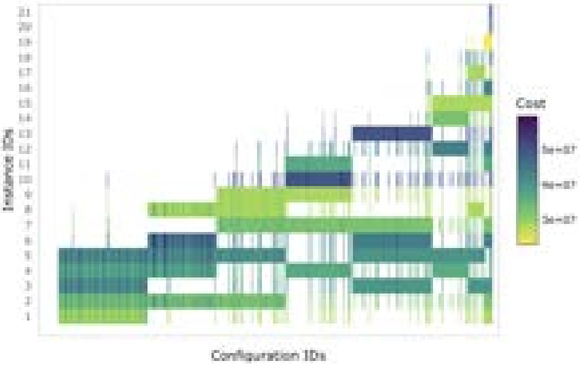
An overview of applications of irace

- Parameter tuning
  - Exact MIP solvers (CPLEX, SCIP [López-Ibáñez & Stützle, 2014])
  - single-objective optimization metaheuristics
  - multi-objective optimization metaheuristics
  - anytime optimization (improve time-quality trade-offs)
  - command-line flags of GCC compiler [Pérez Cáceres et al., 2017]
- Automatic algorithm design
  - From a flexible framework of algorithm components
  - From a grammar description[Martín-Santamaría et al., 2024]
- Machine learning
  - Automatic model selection for survival analysis [Lang et al., 2014]
  - **mlr** R package [Bischl et al., 2013, 2016]
- Design of control software for robots [Francesca et al., 2015]
- Theoretical research [Friedrich et al., 2018; Dang & Doerr, 2019; Hall et al., 2019]

1 919 citations in Google Scholar, 189 000 downloads

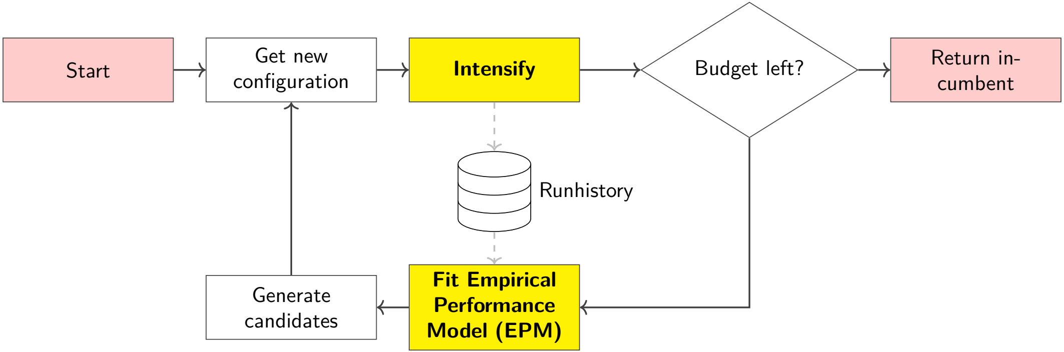
iraceplot: Opening the black-box

https://auto-optimization.github.io/iraceplot/



- Interactive HTML post-configuration report
- Summary statistics per instance / per configuration / per iteration
- Interactive visualizations
- Ablation report

SMAC → MO-SMAC

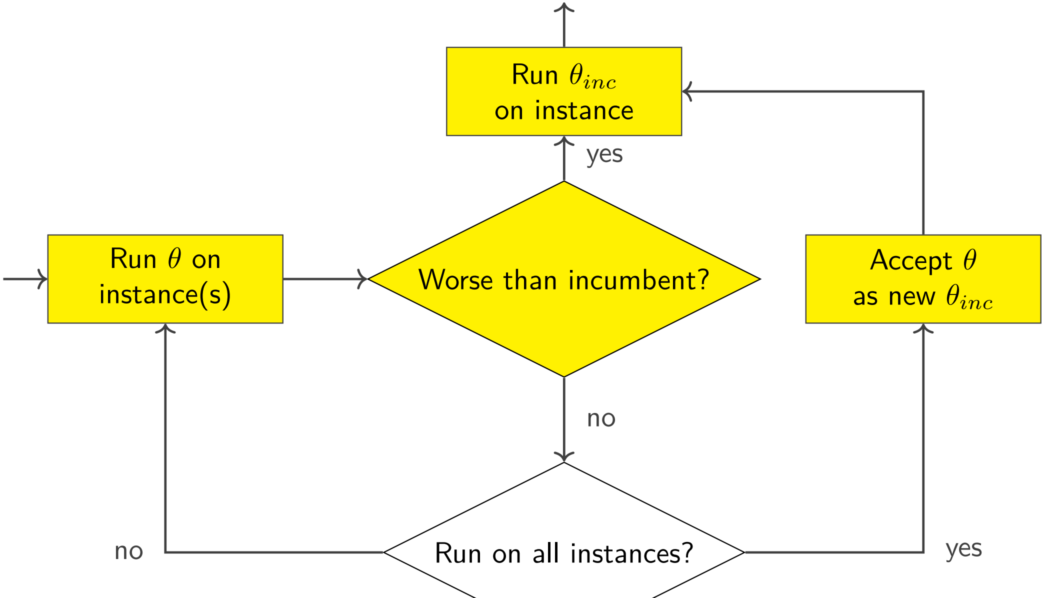


Start
Get new configuration
Intensify
Budget left?
Return in-cumbent
Runhistory
Generate candidates
Fit Empirical Performance Model (EPM)

Modification 1: Intensification

- Incumbent is a *population* of (trade-off) configurations
- Running on instances continues until configuration is closest $\theta_{inc} \in \Theta_{inc}$ *dominates* the challenger
- More configurations in incumbent $\rightarrow$ Less runs on individual incumbent configurations
  - Trade-off!
  - Controlled by new parameter: `max population size`
- θ added to Θ_{inc} if not dominated by any incumbent configuration.
  - Remove incumbent configurations that are

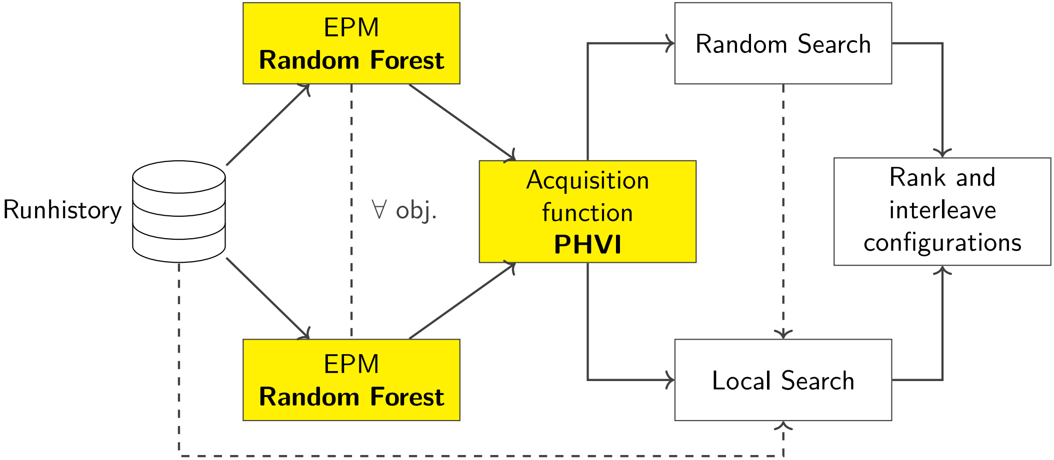
Modification 1: Intensification



Modification 1: Intensification – why the closests?

- Assume a fixed probability p of making a false decision
  - Rejecting a promising configuration
- With an incumbent of size m this probability grows: $1 - (1 - p)^m$
- Hence choose one θ_{inc} from Θ_{inc}

Modification 2: Empirical performance model



Modification 2: Empirical performance model

- single EPM $\rightarrow$ 1 EPM for each objective
- Expected Hypervolume Improvement [Yang et al.,2019]
  - Does not work well with few samples
  - Expensive to compute in > 3 dimensions

Modification 2: Empirical performance model

- single EPM $\rightarrow$ 1 EPM for each objective
- Expected Hypervolume Improvement [Yang et al.,2019]
  - Does not work well with few samples
  - Expensive to compute in > 3 dimensions
- **Predicted Hypervolume Improvement (PHVI)**