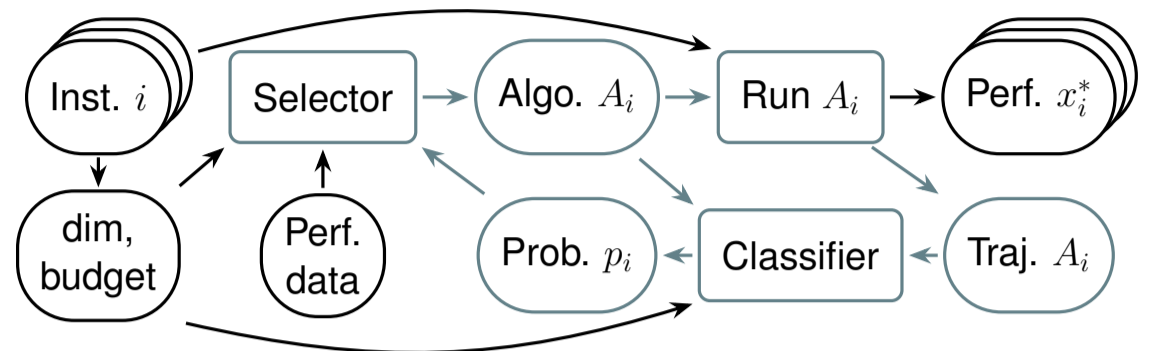


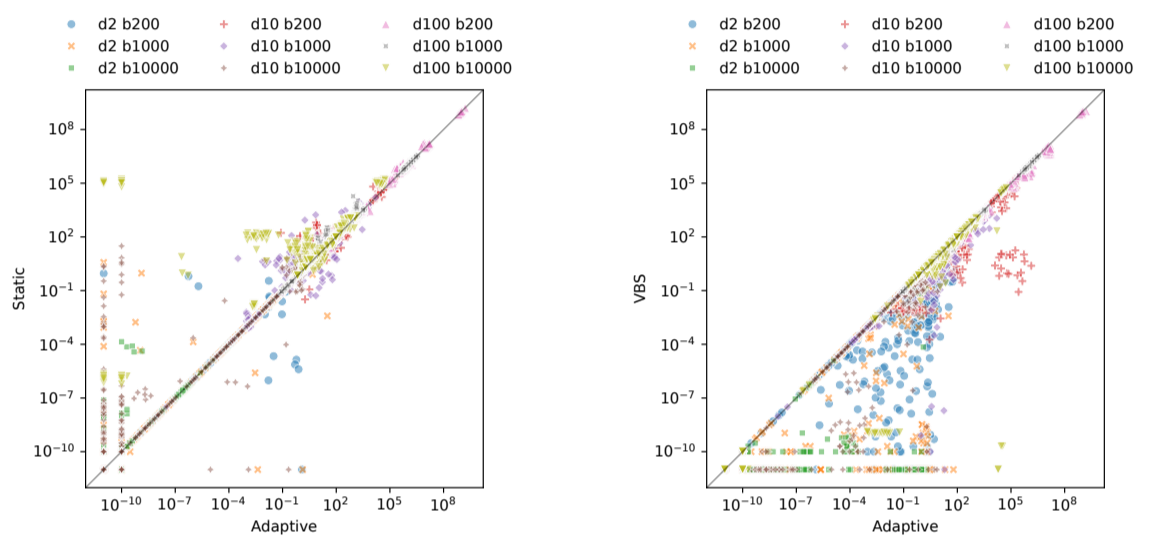
# Specialise algorithm selectors for recurring black-box problems! Cheap, low risk!

## Specialising Algorithm Selectors Over Time for Recurring Problems by Classifying Best-so-far Performance Trajectories

Koen van der Blom,<sup>1</sup> Vanessa Volz,<sup>1</sup> Carola Doerr<sup>2</sup>  
<sup>1</sup>Centrum Wiskunde & Informatica, <sup>2</sup>Sorbonne Université, CNRS



High level overview of the system



Performance vs. static

Performance vs. VBS

## Summary

### Setup

- **Setting:** Recurring black-box problems w/o feature computation
- **Features:** Budget, dimensionality (available w/o evaluations)
- **Train:** Broad set of problems

### After deployment

- **Single recurring problem:** Stream of instances
- **Identify:** Predict problem probabilities from run-trajectories
- **Update:** Use probabilities to adapt selector algorithm choice

### Results

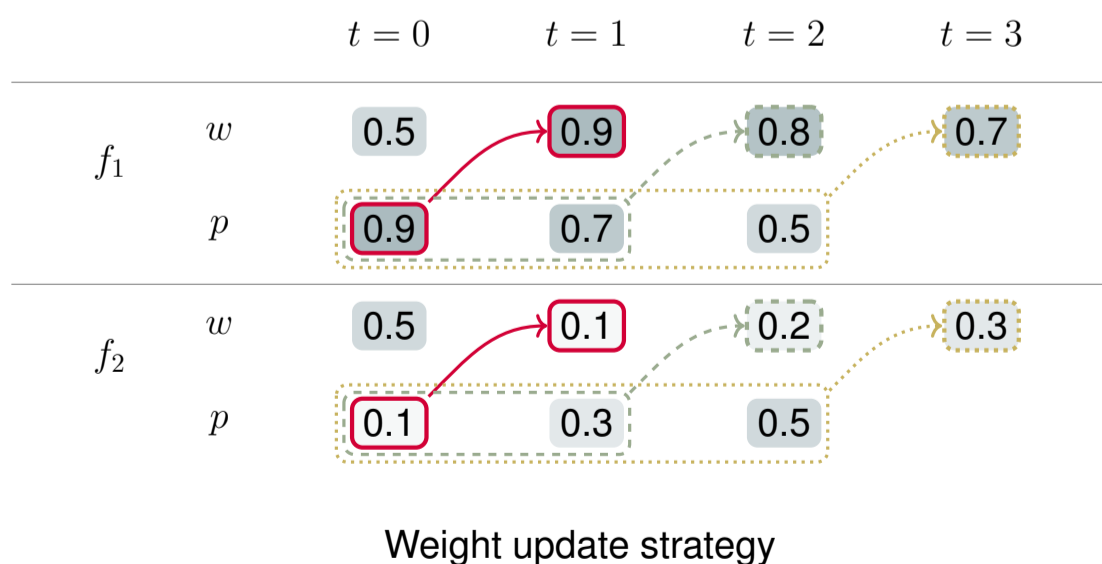
- Median performance never significantly worse than static AS
- Benefits especially for higher dimensional problems
- Selector updates in less than a second

## Wilcoxon test results for the median

Specialised vs.	Dims.	2			10			100		
		Budget	200	1000	10000	200	1000	10000	200	1000
Static	Win	0	0	0	2	3	3	7	7	13
	Draw	24	24	24	22	21	21	17	17	11
	Loss	0	0	0	0	0	0	0	0	0
VBS	Draw	9	18	17	4	13	12	15	21	14
	Loss	15	6	7	20	11	12	9	3	10

## Online prediction of problem probabilities

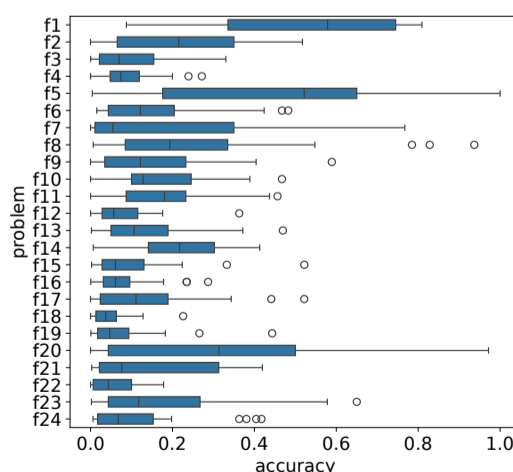
- **Start:** Equal weights = probabilities for each problem
- **After each instance:** Predict problem
- **Scale weights:** Average of predicted probabilities so far



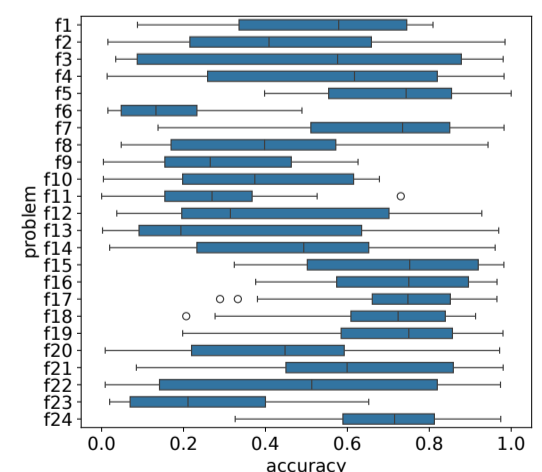
Weight update strategy

## Classification accuracy

- Problem accuracy is not great, but SBS accuracy is good enough
- **Problem accuracy:** Probability assigned to true problem
- **SBS accuracy:** Probability assigned to problem with same SBS



Problem accuracy is bad



SBS accuracy is good

## Experimental details

- **Problem set:** One run for 25 instances for all 24 BBOB functions
- **Train:** 15 instances, **specialise:** 5 instances, **test:** 5 instances
- **Buds.** 200, 1000, 10 000 evaluations; **Dims.**  $d = 2, 10, 100$
- **Algorithms:** Result plots for portfolio size 6, also tested with 34
- **Classifier:** Time series forest, one per algo.-bud.-dim. combo.

## Next steps

- Cluster training instances instead of using pre-defined problems
- Test current-value trajectories to see if they are more informative
- Investigate performance under various other drift scenarios