CAVE:
Configuration, Assessment, Visualization and Evaluation

André Biedenkapp, Joshua Marben,
Marius Lindauer, and Frank Hutter

University of Freiburg

Lion12

- Algorithm parameters can greatly influence an algorithms performance

- Algorithm parameters can greatly influence an algorithms performance
- Success of algorithm configuration:

Domain	#P	Speedup up to	
ASP (*Clasp*)	99	14x	[Gebser et al., 2011]
AI planning (*LPG*)	66	40x	[Vallati et al., 2013]
MIP (*CPLEX*)	76	52x	[Hutter et al., 2010]
SAT (*probSAT*)	9	1500x	[Hutter et al., 2017]

- Algorithm parameters can greatly influence an algorithms performance
- Success of algorithm configuration:

Domain	#P	Speedup up to	
ASP (*Clasp*)	99	14x	[Gebser et al., 2011]
AI planning (*LPG*)	66	40x	[Vallati et al., 2013]
MIP (*CPLEX*)	76	52x	[Hutter et al., 2010]
SAT (*probSAT*)	9	1500x	[Hutter et al., 2017]

- Research focuses on proposing better configuration procedures

Introduction

- Algorithm parameters can greatly influence an algorithms performance
- Success of algorithm configuration:

Domain	#P	Speedup up to	
ASP (*Clasp*)	99	14x	[Gebser et al., 2011]
AI planning (*LPG*)	66	40x	[Vallati et al., 2013]
MIP (*CPLEX*)	76	52x	[Hutter et al., 2010]
SAT (*probSAT*)	9	1500x	[Hutter et al., 2017]

- Research focuses on proposing better configuration procedures
- Resulting procedures only communicate promising parameter settings

Introduction

- Algorithm parameters can greatly influence an algorithms performance
- Success of algorithm configuration:

Domain	#P	Speedup up to	
ASP (*Clasp*)	99	14x	[Gebser et al., 2011]
AI planning (*LPG*)	66	40x	[Vallati et al., 2013]
MIP (*CPLEX*)	76	52x	[Hutter et al., 2010]
SAT (*probSAT*)	9	1500x	[Hutter et al., 2017]

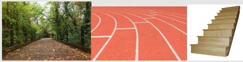
- Research focuses on proposing better configuration procedures
- Resulting procedures only communicate promising parameter settings
- No communication what happened during configuration

Motivation

Algorithm:

Parameter Space:

Instance Set:



Configurator's Input

Configuration
(e.g., SMAC)

Run with best incumbent	run_3	Budget spent evaluating configurations	NaN
# Train instances	302	Cutoff	300
# Test instances	302	Walltime budget	172800
# Parameters	26	Runcount budget	inf
# Features	138	CPU budget	inf
# Duplicate Feature vectors	14	Deterministic	False
		# Runs per Config (min)	0
# Evaluated Configurations	2922	# Runs per Config (mean)	23.8494
# Default evaluations	7	# Runs per Config (max)	2000
# Incumbent evaluations	2000	Total number of configuration runs	69688

Configurator's Output
CAVE's Input

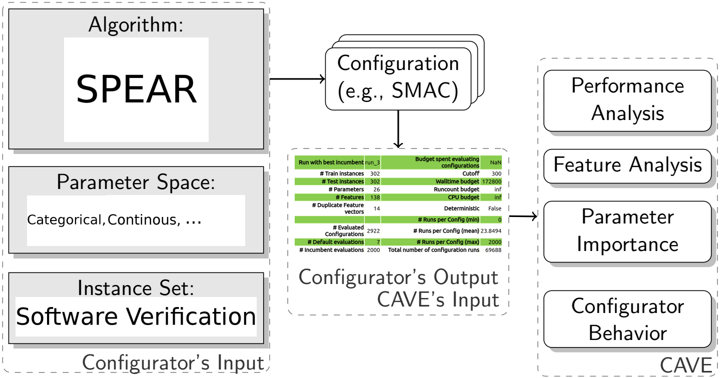
Performance Analysis

Feature Analysis

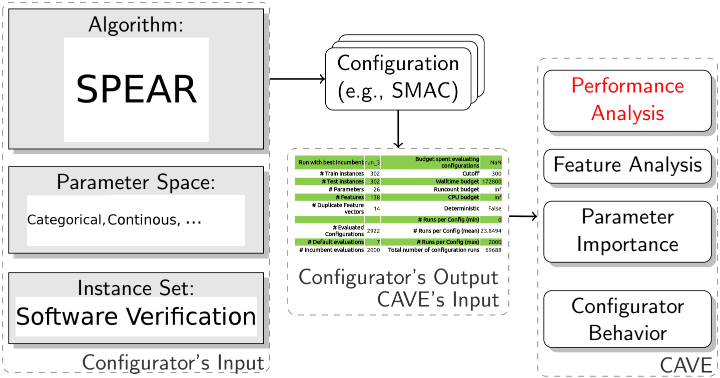
Parameter Importance

Configurator Behavior

CAVE



Run with best incumbent	run_3	Budget spent evaluating configurations	NaN
# Train instances	302	Cutoff	300
# Test instances	302	Walltime budget	172800
# Parameters	26	Runcount budget	inf
# Features	138	CPU budget	inf
# Duplicate Feature vectors	14	Deterministic	False
		# Runs per Config (min)	0
# Evaluated Configurations	2922	# Runs per Config (mean)	23.8494
# Default evaluations	7	# Runs per Config (max)	2000
# Incumbent evaluations	2000	Total number of configuration runs	69688



Run with best incumbent	run_3	Budget spent evaluating configurations	NaN
# Train instances	302	Cutoff	300
# Test instances	302	Walltime budget	172800
# Parameters	26	Runcount budget	inf
# Features	138	CPU budget	inf
# Duplicate Feature vectors	14	Deterministic	False
		# Runs per Config (min)	0
# Evaluated Configurations	2922	# Runs per Config (mean)	23.8494
# Default evaluations	7	# Runs per Config (max)	2000
# Incumbent evaluations	2000	Total number of configuration runs	69688

Performance Analysis (Most Basic)

	Train		Test	
	Default	Incumbent	Default	Incumbent
PAR10				
PAR1				
Timeouts				

Performance Analysis (Most Basic)

	Train		Test	
	Default	Incumbent	Default	Incumbent
PAR10	659.968	11.295	608.726	3.04
PAR1				
Timeouts				

Performance Analysis (Most Basic)

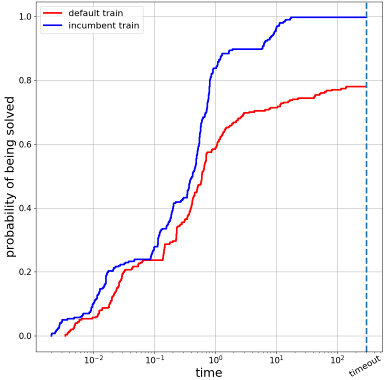
	Train		Test	
	Default	Incumbent	Default	Incumbent
PAR10	659.968	11.295	608.726	3.04
PAR1	69.902	2.355	63.362	3.04
Timeouts				

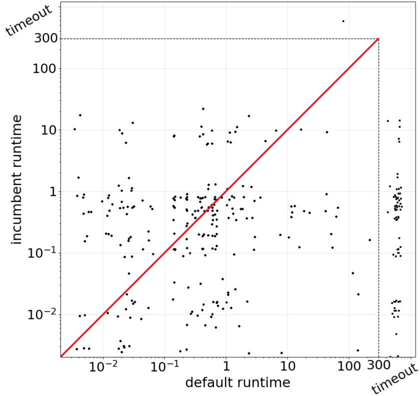
Performance Analysis (Most Basic)

	Train		Test	
	Default	Incumbent	Default	Incumbent
PAR10	659.968	11.295	608.726	3.04
PAR1	69.902	2.355	63.362	3.04
Timeouts	62/302	1/302	55/302	0/302

Performance Analysis (Most Basic)

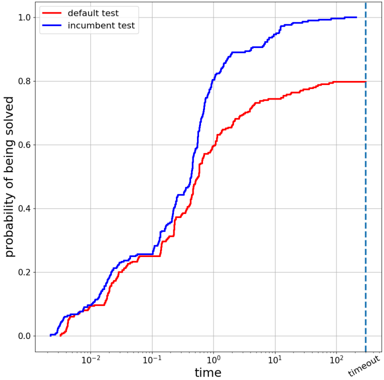
	Train		Test	
	Default	Incumbent	Default	Incumbent
PAR10	659.968	11.295	608.726	3.04
PAR1	69.902	2.355	63.362	3.04
Timeouts	62/302	1/302	55/302	0/302

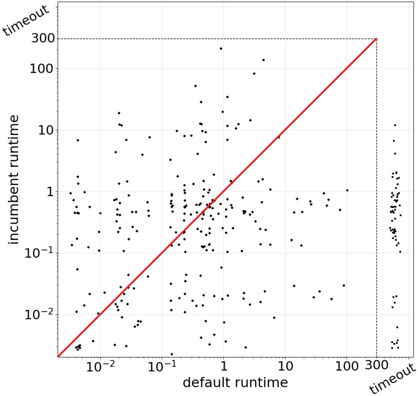




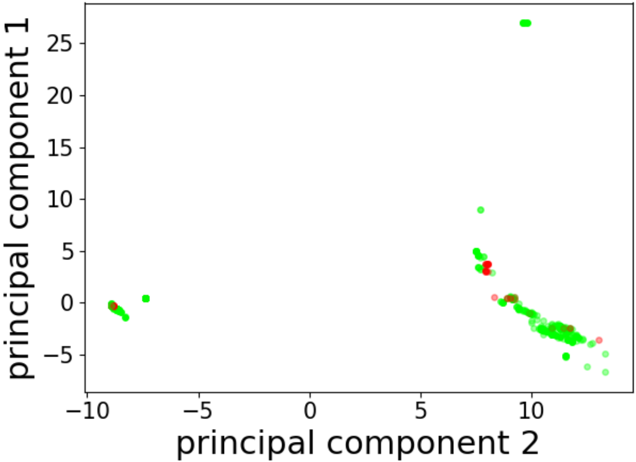
Performance Analysis (Most Basic)

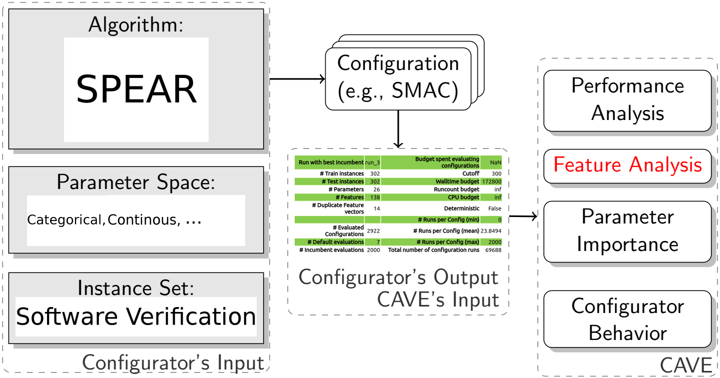
	Train		Test	
	Default	Incumbent	Default	Incumbent
PAR10	659.968	11.295	608.726	3.04
PAR1	69.902	2.355	63.362	3.04
Timeouts	62/302	1/302	55/302	0/302





Algorithm Footprints [Smith-Miles et al., 2014]



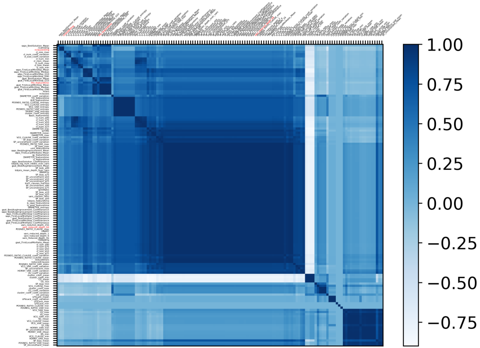


Run with best incumbent	run_3	Budget spent evaluating configurations	NaN
# Train instances	302	Cutoff	300
# Test instances	302	Walltime budget	172800
# Parameters	26	Runcount budget	inf
# Features	138	CPU budget	inf
# Duplicate Feature vectors	14	Deterministic	False
		# Runs per Config (min)	0
# Evaluated Configurations	2922	# Runs per Config (mean)	23.8494
# Default evaluations	7	# Runs per Config (max)	2000
# Incumbent evaluations	2000	Total number of configuration runs	69688

Feature Analysis

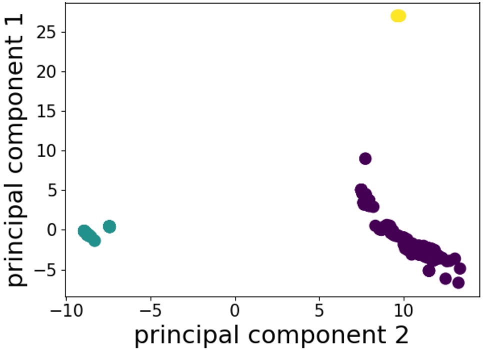
- Instances are characterized by instance features
- Used the feature generator from *SATzilla* [Xu et al., 2008, Hutter et al., 2014]
- $\Rightarrow$ 138 features per instance

- Instances are characterized by instance features
- Used the feature generator from *SATzilla* [Xu et al., 2008, Hutter et al., 2014]
- ⇒ 138 features per instance
- Feature Correlation



Feature Analysis

- Instances are characterized by instance features
- Used the feature generator from *SATzilla* [Xu et al., 2008, Hutter et al., 2014]
- $\Rightarrow$ 138 features per instance
- Clustering



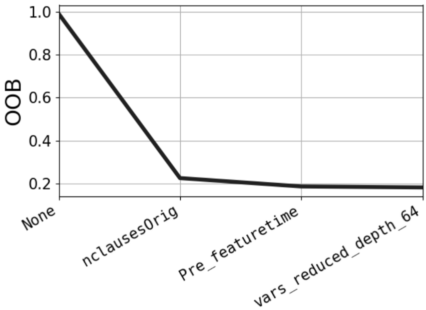
Feature Analysis

- Instances are characterized by instance features
- Used the feature generator from *SATzilla* [Xu et al., 2008, Hutter et al., 2014]
- ⇒ 138 features per instance
- Feature importance based on greedy forward selection [Hutter et al., 2013]

	Error
None	0.989727
nclausesOrig	0.225080
Pre_featuretime	0.186257
vars_reduced_depth_64	0.181692

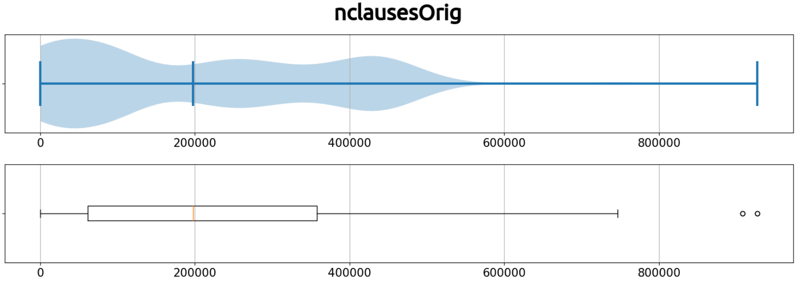
Feature Analysis

- Instances are characterized by instance features
- Used the feature generator from *SATzilla* [Xu et al., 2008, Hutter et al., 2014]
- $\Rightarrow$ 138 features per instance
- Feature importance based on greedy forward selection [Hutter et al., 2013]

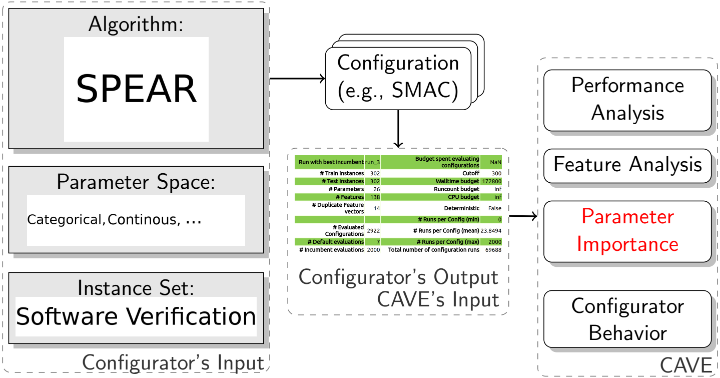


Feature Analysis

- Instances are characterized by instance features
- Used the feature generator from *SATzilla* [Xu et al., 2008, Hutter et al., 2014]
- $\Rightarrow$ 138 features per instance
- Box and violin plots for each feature



Parameter Importance



Run with best incumbent	run_3	Budget spent evaluating configurations	NaN
# Train instances	302	Cutoff	300
# Test instances	302	Walltime budget	172800
# Parameters	26	Runcount budget	inf
# Features	138	CPU budget	inf
# Duplicate Feature vectors	14	Deterministic	False
		# Runs per Config (min)	0
# Evaluated Configurations	2922	# Runs per Config (mean)	23.8494
# Default evaluations	7	# Runs per Config (max)	2000
# Incumbent evaluations	2000	Total number of configuration runs	69688

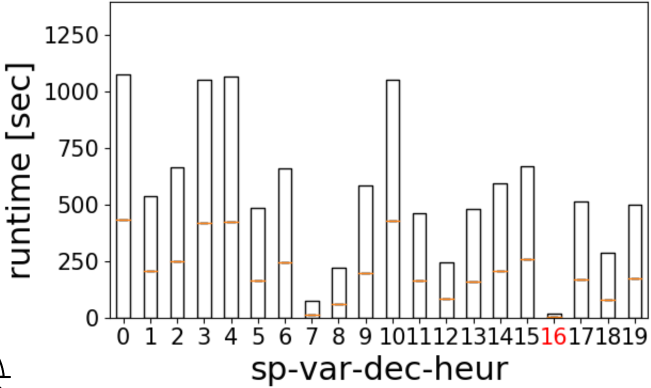
	fANOVA	Ablation	LPI
sp-var-dec-heur	65.06	73.90	91.36
sp-orig-clause-sort-heur	1.31	21.94	-
sp-phase-dec-heur	5.94	-	-
sp-restart-inc	-	1.44	4.05
sp-first-restart	-	-	1.59
sp-learned-clause-sort-heur	1.12	2.02	-
sp-variable-decay	-	-	1.50

CAVE: Local Parameter Importance (LPI)

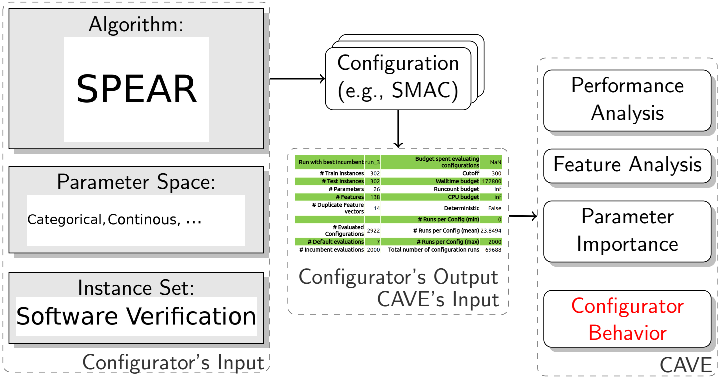
- Novel importance analysis method
- Inspired by the human strategy to look much performance of configurations in the neighborhood of incumbent degrades
- Uses empirical performance model to predict performance of neighboring configurations [Biedenkapp et al., 2017]

CAVE: Local Parameter Importance (LPI)

- Novel importance analysis method
- Inspired by the human strategy to look much performance of configurations in the neighborhood of incumbent degrades
- Uses empirical performance model to predict performance of neighboring configurations [Biedenkapp et al., 2017]

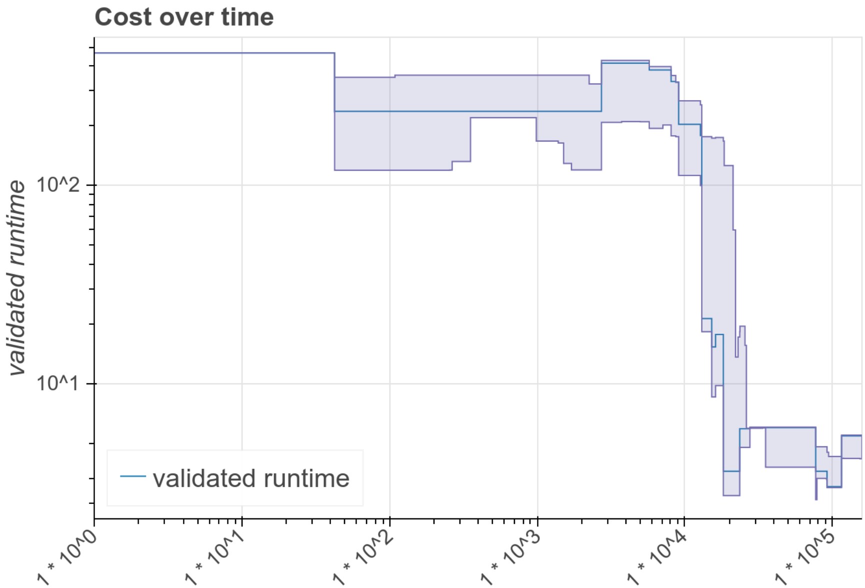


Configurator Behaviour

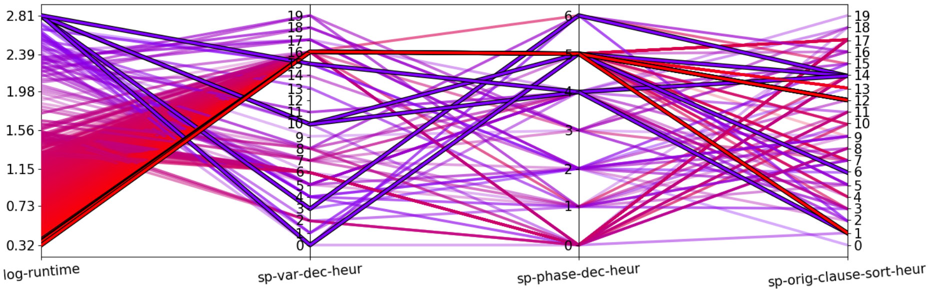


Run with best incumbent	run_3	Budget spent evaluating configurations	NaN
# Train instances	302	Cutoff	300
# Test instances	302	Walltime budget	172800
# Parameters	26	Runcount budget	inf
# Features	138	CPU budget	inf
# Duplicate Feature vectors	14	Deterministic	False
		# Runs per Config (min)	0
# Evaluated Configurations	2922	# Runs per Config (mean)	23.8494
# Default evaluations	7	# Runs per Config (max)	2000
# Incumbent evaluations	2000	Total number of configuration runs	69688

CAVE: Configurator Behavior



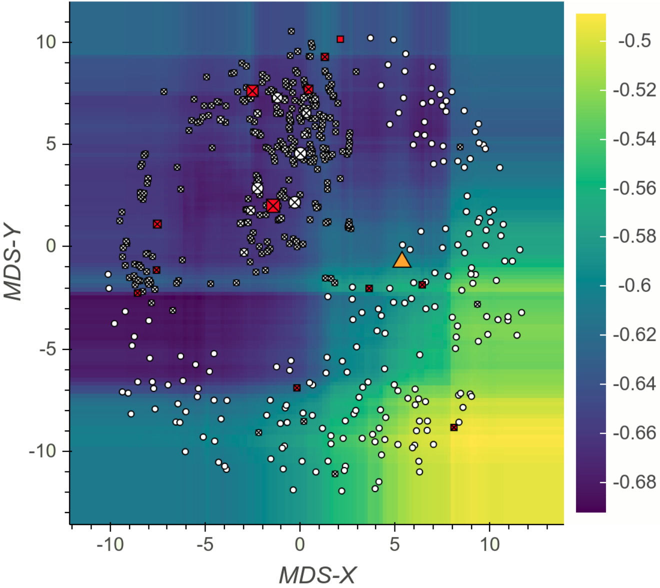
Parallel Coordinates [Heinrich and Weiskopf, 2013]



Configurator Footprint

based on similarity metric by [Xu et al., 2016]

$\frac{1}{10}$ budget spent

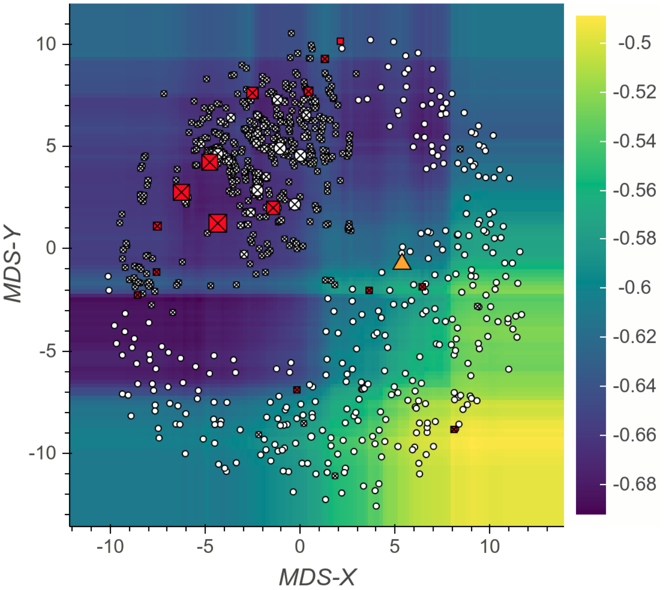


○ Configuration (Random)
⊗ Configuration (Acquisition)
■ Incumbent on trajectory
▼ Final Incumbent
△ Default

Configurator Footprint

based on similarity metric by [Xu et al., 2016]

$\frac{2}{10}$ budget spent

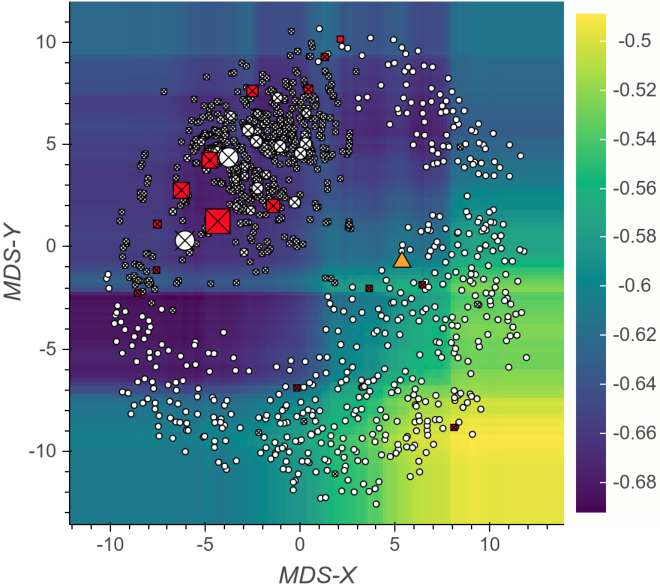


○ Configuration (Random)
⊗ Configuration (Acquisition)
■ Incumbent on trajectory
▼ Final Incumbent
▲ Default

Configurator Footprint
based on similarity metric by [Xu et al., 2016]

$\frac{3}{10}$ budget spent

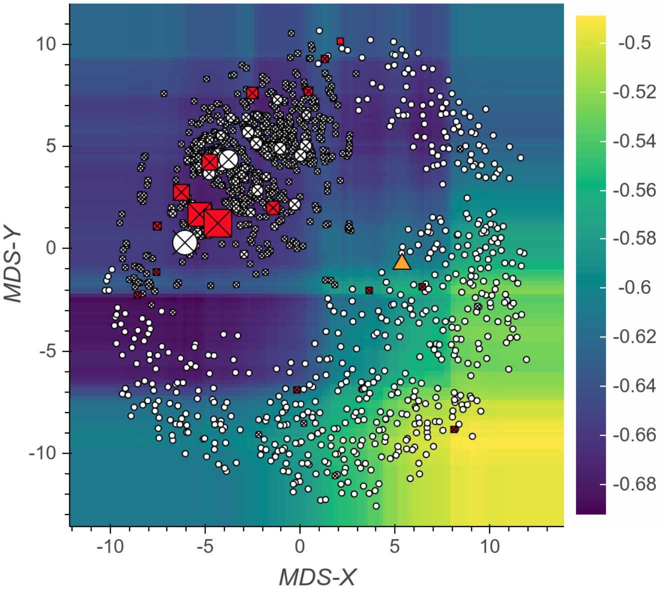
○ Configuration (Random)
⊗ Configuration (Acquisition)
■ Incumbent on trajectory
▼ Final Incumbent
▲ Default



Configurator Footprint
based on similarity metric by [Xu et al., 2016]

$\frac{4}{10}$ budget spent

○ Configuration (Random)
⊗ Configuration (Acquisition)
■ Incumbent on trajectory
▼ Final Incumbent
△ Default

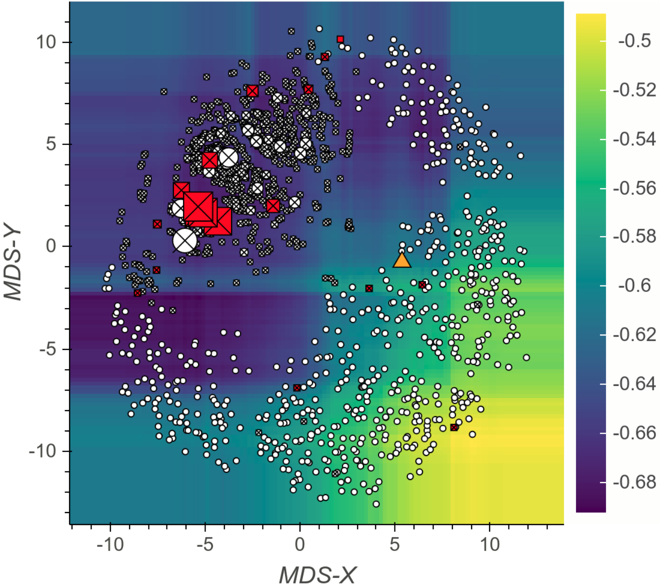


Configurator Footprint

based on similarity metric by [Xu et al., 2016]

$\frac{5}{10}$ budget spent

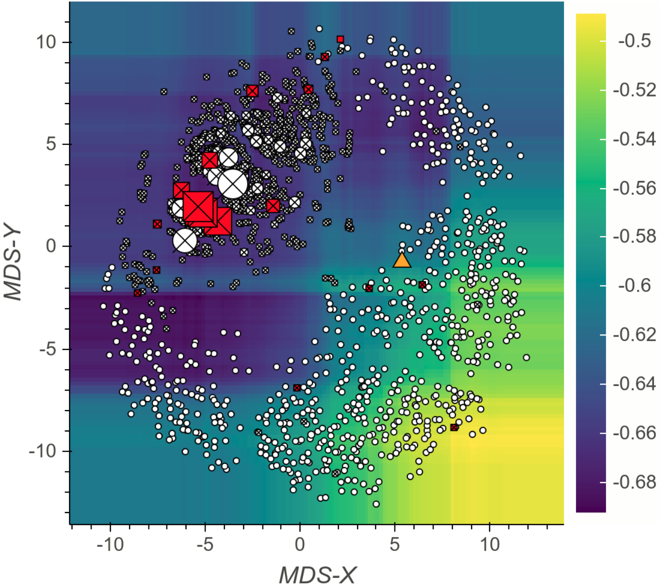
○ Configuration (Random)
⊗ Configuration (Acquisition)
■ Incumbent on trajectory
▼ Final Incumbent
▲ Default



Configurator Footprint
based on similarity metric by [Xu et al., 2016]

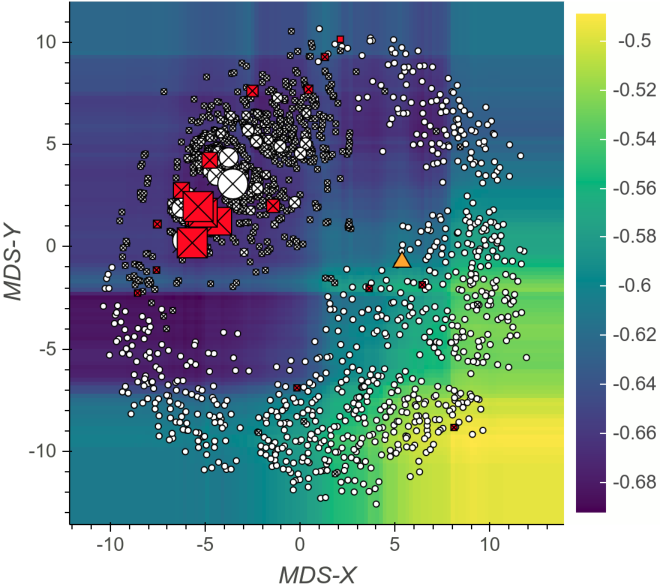
$\frac{6}{10}$ budget spent

○ Configuration (Random)
⊗ Configuration (Acquisition)
■ Incumbent on trajectory
▼ Final Incumbent
▲ Default



Configurator Footprint
based on similarity metric by [Xu et al., 2016]

$\frac{7}{10}$ budget spent

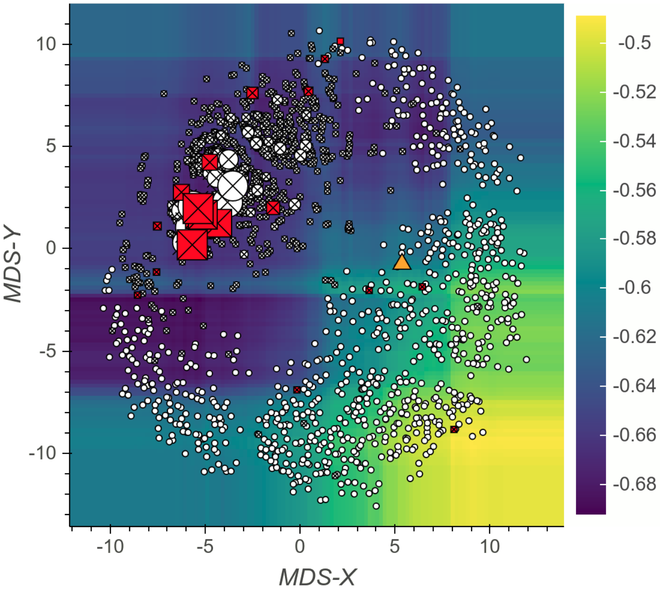


Configurator Footprint

based on similarity metric by [Xu et al., 2016]

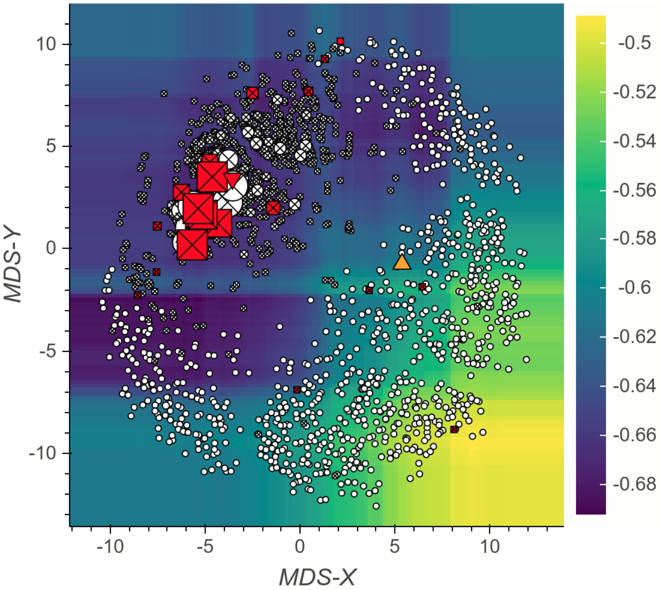
$\frac{8}{10}$ budget spent

○ Configuration (Random)
⊗ Configuration (Acquisition)
■ Incumbent on trajectory
▼ Final Incumbent
▲ Default



Configurator Footprint
based on similarity metric by [Xu et al., 2016]

$\frac{9}{10}$ budget spent

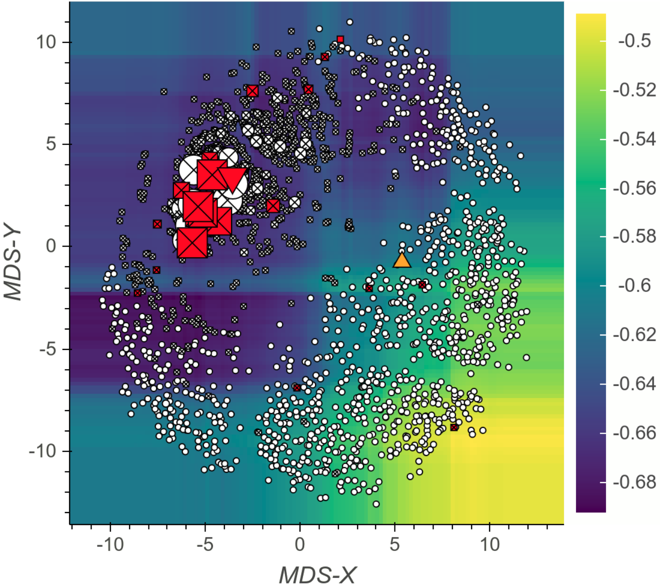


○ Configuration (Random)
⊗ Configuration (Acquisition)
■ Incumbent on trajectory
▼ Final Incumbent
▲ Default

Configurator Footprint

based on similarity metric by [Xu et al., 2016]

$\frac{10}{10}$ budget spent



○ Configuration (Random)
⊗ Configuration (Acquisition)
■ Incumbent on trajectory
▼ Final Incumbent
▲ Default

Q1 Does the set of important parameters change depending on the instance set?

Q2 Do local and global parameter importance approaches agree on the set of important parameters?

CAVE: Case Study

Algorithm	Domain	#P	#Insts.
LPG[Gerevini and Serina, 2002]	AI plan.	65	3
Clasp(-ASP)[Gebser et al., 2012]	ASP	98	3
CPLEX	MIP	74	4
SATenstein[KhudaBukhsh et al., 2009]	SAT	49	6
Clasp(-HAND)	SAT	75	3
Clasp(-RAND)	SAT	75	3
probSAT[Balint and Schöning, 2012]	SAT	9	3

CAVE: Case Study

Algorithm	ablation μ	fANOVA μ	LPI μ
clasp(-ASP)	$\approx$ 8%	$\approx$ 42%	$\approx$ 31%

CAVE: Case Study

Algorithm	ablation μ	fANOVA μ	LPI μ
clasp(-ASP)	$\approx$ 8%	$\approx$ 42%	$\approx$ 31%
clasp(-HAND)	$\approx$ 0%	$\approx$ 50%	$\approx$ 25%
clasp(-RAND)	$\approx$ 14%	$\approx$ 11%	$\approx$ 28%

CAVE: Case Study

Algorithm	ablation μ	fANOVA μ	LPI μ
clasp(-ASP)	$\approx$ 8%	$\approx$ 42%	$\approx$ 31%
clasp(-HAND)	$\approx$ 0%	$\approx$ 50%	$\approx$ 25%
clasp(-RAND)	$\approx$ 14%	$\approx$ 11%	$\approx$ 28%
CPLEX	$\approx$ 4%	$\approx$ 16%	$\approx$ 36%

Cave: Case Study

Algorithm	ablation μ	fANOVA μ	LPI μ
clasp(-ASP)	$\approx$ 8%	$\approx$ 42%	$\approx$ 31%
clasp(-HAND)	$\approx$ 0%	$\approx$ 50%	$\approx$ 25%
clasp(-RAND)	$\approx$ 14%	$\approx$ 11%	$\approx$ 28%
CPLEX	$\approx$ 4%	$\approx$ 16%	$\approx$ 36%
lpg	$\approx$ 16%	$\approx$ 30%	$\approx$ 38%

CAVE: Case Study

Algorithm	ablation μ	fANOVA μ	LPI μ
clasp(-ASP)	$\approx$ 8%	$\approx$ 42%	$\approx$ 31%
clasp(-HAND)	$\approx$ 0%	$\approx$ 50%	$\approx$ 25%
clasp(-RAND)	$\approx$ 14%	$\approx$ 11%	$\approx$ 28%
CPLEX	$\approx$ 4%	$\approx$ 16%	$\approx$ 36%
lpg	$\approx$ 16%	$\approx$ 30%	$\approx$ 38%
probSAT	$\approx$ 47%	$\approx$ 32%	$\approx$ 61%

CAVE: Case Study

Algorithm	ablation μ	fANOVA μ	LPI μ
clasp(-ASP)	$\approx$ 8%	$\approx$ 42%	$\approx$ 31%
clasp(-HAND)	$\approx$ 0%	$\approx$ 50%	$\approx$ 25%
clasp(-RAND)	$\approx$ 14%	$\approx$ 11%	$\approx$ 28%
CPLEX	$\approx$ 4%	$\approx$ 16%	$\approx$ 36%
lpg	$\approx$ 16%	$\approx$ 30%	$\approx$ 38%
probSAT	$\approx$ 47%	$\approx$ 32%	$\approx$ 61%
SATenstein	$\approx$ 15%	$\approx$ 26%	$\approx$ 27%

- $\Rightarrow$ parameter importance depends on the instance set

CAVE: Case Study

Algorithm	ablation μ	fANOVA μ	LPI μ
clasp(-ASP)	$\approx$ 8%	$\approx$ 42%	$\approx$ 31%
clasp(-HAND)	$\approx$ 0%	$\approx$ 50%	$\approx$ 25%
clasp(-RAND)	$\approx$ 14%	$\approx$ 11%	$\approx$ 28%
CPLEX	$\approx$ 4%	$\approx$ 16%	$\approx$ 36%
lpg	$\approx$ 16%	$\approx$ 30%	$\approx$ 38%
probSAT	$\approx$ 47%	$\approx$ 32%	$\approx$ 61%
SATenstein	$\approx$ 15%	$\approx$ 26%	$\approx$ 27%

- $\Rightarrow$ parameter importance depends on the instance set
- A subset of parameters is important across instance sets

CAVE: Case Study

Algorithm	fANOVA vs. ablation μ	fANOVA vs. LPI μ	ablation vs. LPI μ
clasp(-ASP)	$\approx$ 8%	$\approx$ 6%	$\approx$ 12%

CAVE: Case Study

Algorithm	fANOVA vs. ablation μ	fANOVA vs. LPI μ	ablation vs. LPI μ
clasp(-ASP)	$\approx$ 8%	$\approx$ 6%	$\approx$ 12%
clasp(-HAND)	$\approx$ 7%	$\approx$ 10%	$\approx$ 22%
clasp(-RAND)	$\approx$ 38%	$\approx$ 13%	$\approx$ 32%

CAVE: Case Study

Algorithm	fANOVA vs. ablation μ	fANOVA vs. LPI μ	ablation vs. LPI μ
clasp(-ASP)	$\approx$ 8%	$\approx$ 6%	$\approx$ 12%
clasp(-HAND)	$\approx$ 7%	$\approx$ 10%	$\approx$ 22%
clasp(-RAND)	$\approx$ 38%	$\approx$ 13%	$\approx$ 32%
CPLEX	$\approx$ 7%	$\approx$ 7%	$\approx$ 13%

ML4AAD

CAVE: Case Study

Algorithm	fANOVA vs. ablation μ	fANOVA vs. LPI μ	ablation vs. LPI μ
clasp(-ASP)	$\approx$ 8%	$\approx$ 6%	$\approx$ 12%
clasp(-HAND)	$\approx$ 7%	$\approx$ 10%	$\approx$ 22%
clasp(-RAND)	$\approx$ 38%	$\approx$ 13%	$\approx$ 32%
CPLEX	$\approx$ 7%	$\approx$ 7%	$\approx$ 13%
lpg	$\approx$ 43%	$\approx$ 38%	$\approx$ 39%

CAVE: Case Study

Algorithm	fANOVA vs. ablation μ	fANOVA vs. LPI μ	ablation vs. LPI μ
clasp(-ASP)	$\approx$ 8%	$\approx$ 6%	$\approx$ 12%
clasp(-HAND)	$\approx$ 7%	$\approx$ 10%	$\approx$ 22%
clasp(-RAND)	$\approx$ 38%	$\approx$ 13%	$\approx$ 32%
CPLEX	$\approx$ 7%	$\approx$ 7%	$\approx$ 13%
lpg	$\approx$ 43%	$\approx$ 38%	$\approx$ 39%
probSAT	$\approx$ 4%	$\approx$ 22%	$\approx$ 32%

CAVE: Case Study

Algorithm	fANOVA vs. ablation μ	fANOVA vs. LPI μ	ablation vs. LPI μ
clasp(-ASP)	$\approx$ 8%	$\approx$ 6%	$\approx$ 12%
clasp(-HAND)	$\approx$ 7%	$\approx$ 10%	$\approx$ 22%
clasp(-RAND)	$\approx$ 38%	$\approx$ 13%	$\approx$ 32%
CPLEX	$\approx$ 7%	$\approx$ 7%	$\approx$ 13%
lpg	$\approx$ 43%	$\approx$ 38%	$\approx$ 39%
probSAT	$\approx$ 4%	$\approx$ 22%	$\approx$ 32%
SATenstein	$\approx$ 12%	$\approx$ 13%	$\approx$ 34%

- *fANOVA* and *ablation* tend to view different parameters as important

CAVE: Case Study

Algorithm	fANOVA vs. ablation μ	fANOVA vs. LPI μ	ablation vs. LPI μ
clasp(-ASP)	$\approx$ 8%	$\approx$ 6%	$\approx$ 12%
clasp(-HAND)	$\approx$ 7%	$\approx$ 10%	$\approx$ 22%
clasp(-RAND)	$\approx$ 38%	$\approx$ 13%	$\approx$ 32%
CPLEX	$\approx$ 7%	$\approx$ 7%	$\approx$ 13%
lpg	$\approx$ 43%	$\approx$ 38%	$\approx$ 39%
probSAT	$\approx$ 4%	$\approx$ 22%	$\approx$ 32%
SATenstein	$\approx$ 12%	$\approx$ 13%	$\approx$ 34%

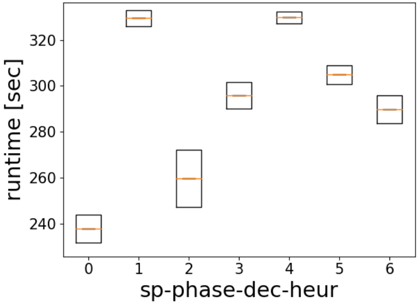
- *fANOVA* and *ablation* tend to view different parameters as important
- $\Rightarrow$ global and local parameter importance give different view on parameter importance

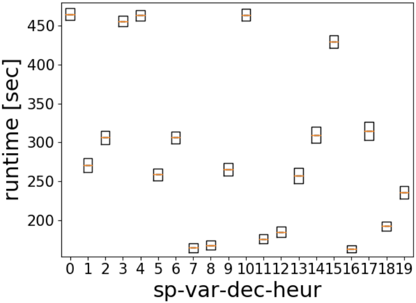
Wrap-up

- Presented automatic analysis tool
- Introduced two new analysis approaches
  - Local Parameter Importance
  - Configurator Footprints
- Demonstrated the usefulness of this tool by demonstrating
  - different analysis approaches on a running example
  - Parameter importance depends on the examined instance set
  - Global and local importance analysis are complementary

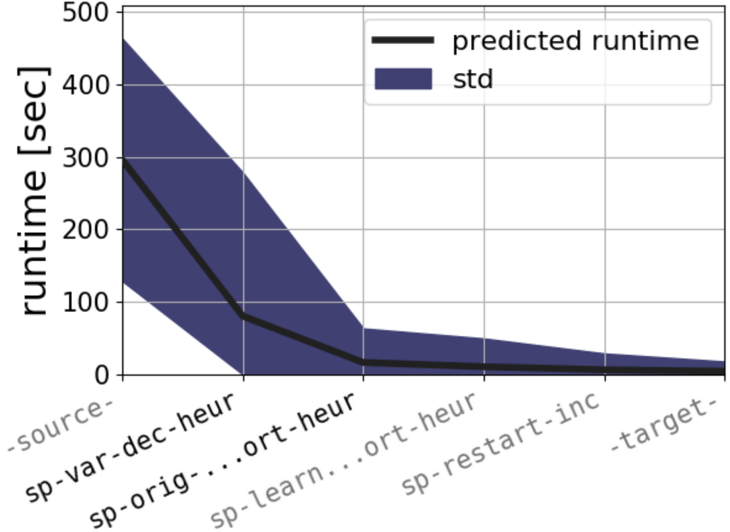
Full report

http://ml.informatik.uni-freiburg.de/~biedenka/cave.html


runtime [sec]
320
300
280
260
240
0 1 2 3 4 5 6
sp-phase-dec-heur


runtime [sec]
450
400
350
300
250
200
0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19
sp-var-dec-heur

CAVE: *ablation*



Configurator Footprint:

1. For each pair of configurations compute similarity $s(\theta_i, \theta_j)$ [Xu et al., 2016]
2. Fit 2D *MDS* based on similarities
3. Plot each configuration θ in 2D space $MDS(\theta)$, size proportional to evaluations
4. Highlight incumbents of trajectory
5. Fit EPM $\hat{c} : \mathbb{R}^2 \times \Pi \rightarrow \mathbb{R}$ based on runhistory
6. Plot heatmap in background based on marginalized predicted performance

References I

Balint, A. and Schöning, U. (2012).
Choosing probability distributions for stochastic local search and the role of make versus break.
Theory and Applications of Satisfiability Testing–SAT 2012, pages 16–29.

Biedenkapp, A., Lindauer, M., Eggensperger, K., Fawcett, C., Hoos, H., and Hutter, F. (2017).
Efficient parameter importance analysis via ablation with surrogates.
In *Proc. of AAAI'17*, pages 773–779.

Gebser, M., Kaminski, R., Kaufmann, B., Ostrowski, M., Schaub, T., and Schneider, M. (2011).
Potassco: The Potsdam answer set solving collection.
AICom, 24(2):107–124.

Gebser, M., Kaufmann, B., and Schaub, T. (2012).
Conflict-driven answer set solving: From theory to practice.
AI, 187-188:52–89.

Gerevini, A. and Serina, I. (2002).
LPG: A planner based on local search for planning graphs with action costs.
In *Proc. of AIPS'02*, pages 13–22.

Heinrich, J. and Weiskopf, D. (2013).
State of the art of parallel coordinates.
In *Proceedings of Eurographics*, pages 95–116. Eurographics Association.

Hutter, F., Hoos, H., and Leyton-Brown, K. (2010).
Automated configuration of mixed integer programming solvers.
In *Proc. of CPAIOR'10*, pages 186–202.

Hutter, F., Hoos, H., and Leyton-Brown, K. (2013).
Identifying key algorithm parameters and instance features using forward selection.
In *Proc. of LION'13*, pages 364–381.

Hutter, F., Lindauer, M., Balint, A., Bayless, S., Hoos, H., and Leyton-Brown, K. (2017).
The configurable SAT solver challenge (CSSC).
AIJ, 243:1–25.

Hutter, F., Xu, L., Hoos, H., and Leyton-Brown, K. (2014).
Algorithm runtime prediction: Methods and evaluation.
AIJ, 206:79–111.

KhudaBukhsh, A., Xu, L., Hoos, H., and Leyton-Brown, K. (2009).
SATenstein: Automatically building local search SAT solvers from components.
In *Proc. of IJCAI'09*, pages 517–524.

Smith-Miles, K., Baatar, D., Wreford, B., and Lewis, R. (2014).
Towards objective measures of algorithm performance across instance space.
Computers & OR, 45:12–24.

Vallati, M., Fawcett, C., Gerevini, A., Hoos, H., and Saetti, A. (2013).
Automatic generation of efficient domain-optimized planners from generic parametrized planners.

Xu, L., Hutter, F., Hoos, H., and Leyton-Brown, K. (2008).
SATzilla: Portfolio-based algorithm selection for SAT.
JAIR, 32:565–606.

References III

Xu, L., KhudaBukhsh, A., Hoos, H., and Leyton-Brown, K. (2016).
Quantifying the similarity of algorithm configurations.
In *Proc. of LION'16*, pages 203–217.