AutoFolio:
An automatically configured Algorithm Selector

Marius Lindauer\textsuperscript{1} \quad Holger Hoos\textsuperscript{2,3}

Frank Hutter\textsuperscript{1} \quad Torsten Schaub\textsuperscript{4}

\textsuperscript{1} Institute of Computer Science
University of Freiburg
Germany

\textsuperscript{3} Department of Computer Science
University of British Columbia
Canada

\textsuperscript{2} Leiden Institute of Advanced Computer Science
University of Leiden
The Netherlands

\textsuperscript{4} Institute of Computer Science
University of Potsdam
Germany
Traditional approach to solver construction:

- explore variants during development,
Traditional approach to solver construction:

- explore variants during development,
  make design choices based on limited experiments
Traditional approach to solver construction:

- explore variants during development, make design choices based on limited experiments
- deploy & run solver
Traditional approach to solver construction:

- explore variants during development, make design choices based on limited experiments
- deploy & run solver

Programming by Optimisation: [Hoos 2012]

- explore variants during development,
Traditional approach to solver construction:

- explore variants during development, make design choices based on limited experiments
- deploy & run solver

Programming by Optimisation: [Hoos 2012]

- explore variants during development, leave design choices open \(\rightsquigarrow\) (large) space of solvers
Traditional approach to solver construction:

- explore variants during development,
  make design choices based on limited experiments

- deploy & run solver

Programming by Optimisation: [Hoos 2012]

- explore variants during development,
  leave design choices open $\leadsto$ (large) space of solvers

- automatically determine performance-optimised design for given use context
Per-Instance Algorithm Selection [Rice 1976]

- Instance
  - Compute Features
  - Algorithm Portfolio
  - Select Algorithm
  - Solve Instance with Algorithm

1. 3 to 15.4 $\times$ speedup over single best algorithm for SAT, MAXSAT, CSP, QBF, ASP, ...
Per-Instance Algorithm Selection [Rice 1976]

\[ \sim 1.3 \text{ to } 15.4 \times \text{ speedup over single best algorithm for SAT, MAXSAT, CSP, QBF, ASP, \ldots} \]
Many algorithm selectors . . .

- SATzilla’11 [Xu et al. 2011]: cost-sensitive random forest classification; pre-solving schedule
Many algorithm selectors . . .

- **SATzilla’11** [Xu et al. 2011]:
  cost-sensitive random forest classification; pre-solving schedule

- **ISAC** [Kadioglu et al. 2010]:
  $g$-means clustering
Many algorithm selectors . . .

- **SATzilla’11** [Xu et al. 2011]: cost-sensitive random forest classification; pre-solving schedule
- **ISAC** [Kadioglu et al. 2010]: $g$-means clustering
- **3S** [Kadioglu et al. 2011]: $k$-NN; pre-solving schedule
Many algorithm selectors . . .

- **SATzilla’11** [Xu et al. 2011]:
  cost-sensitive random forest classification; pre-solving schedule

- **ISAC** [Kadioglu et al. 2010]:
  $g$-means clustering

- **3S** [Kadioglu et al. 2011]:
  $k$-NN; pre-solving schedule

- **SNNAP** [Collautti et al. 2013]:
  random forest regression + $k$-NN
Many algorithm selectors . . .

- **SATzilla’11** [Xu et al. 2011]:
  cost-sensitive random forest classification; pre-solving schedule

- **ISAC** [Kadioglu et al. 2010]:
  $g$-means clustering

- **3S** [Kadioglu et al. 2011]:
  $k$-NN; pre-solving schedule

- **SNNAP** [Collautti et al. 2013]:
  random forest regression + $k$-NN

- . . .
...but none rules all the others:

**Speedup** against single best algorithm on algorithm selection scenarios from *ASlib* [Bischl *et al.* 2016]
but none rules all the others:

**Speedup** against single best algorithm on algorithm selection scenarios from *ASlib* [Bischl *et al.* 2016]

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>SATzilla’11</th>
<th>3S</th>
<th>aspeed</th>
</tr>
</thead>
<tbody>
<tr>
<td>SAT12-ALL</td>
<td>1.8</td>
<td>1.5</td>
<td>1.1</td>
</tr>
</tbody>
</table>
...but none rules all the others:

**Speedup** against single best algorithm on algorithm selection scenarios from *ASlib* [Bischl et al. 2016]

<table>
<thead>
<tr>
<th></th>
<th>SATzilla’11</th>
<th>3S</th>
<th>aspeed</th>
</tr>
</thead>
<tbody>
<tr>
<td>SAT12-ALL</td>
<td>1.8</td>
<td>1.5</td>
<td>1.1</td>
</tr>
<tr>
<td>ASP-POTASSCO</td>
<td>4.2</td>
<td>4.1</td>
<td>1.4</td>
</tr>
<tr>
<td>CSP-2010</td>
<td>3.1</td>
<td>1.5</td>
<td>1.0</td>
</tr>
<tr>
<td>MAXSAT12-PMS</td>
<td>8.6</td>
<td>6.5</td>
<td>2.7</td>
</tr>
</tbody>
</table>
...but none rules all the others:

**Speedup** against single best algorithm on algorithm selection scenarios from *ASlib* [Bischl et al. 2016]

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>SATzilla’11</th>
<th>3S</th>
<th>aspeed</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>SAT12-ALL</strong></td>
<td><strong>1.8</strong></td>
<td>1.5</td>
<td>1.1</td>
</tr>
<tr>
<td><strong>ASP-POTASSCO</strong></td>
<td><strong>4.2</strong></td>
<td>4.1</td>
<td>1.4</td>
</tr>
<tr>
<td><strong>CSP-2010</strong></td>
<td><strong>3.1</strong></td>
<td>1.5</td>
<td>1.0</td>
</tr>
<tr>
<td><strong>MAXSAT12-PMS</strong></td>
<td><strong>8.6</strong></td>
<td>6.5</td>
<td>2.7</td>
</tr>
<tr>
<td><strong>PREMARSHALLING (OR)</strong></td>
<td><strong>2.3</strong></td>
<td>2.9</td>
<td><strong>3.6</strong></td>
</tr>
</tbody>
</table>
...but none rules all the others:

**Speedup** against single best algorithm on algorithm selection scenarios from *ASlib* [Bischl et al. 2016]

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>SATzilla’11</th>
<th>3S</th>
<th>aspeed</th>
</tr>
</thead>
<tbody>
<tr>
<td>SAT12-ALL</td>
<td>1.8</td>
<td>1.5</td>
<td>1.1</td>
</tr>
<tr>
<td>ASP-POTASSCO</td>
<td>4.2</td>
<td>4.1</td>
<td>1.4</td>
</tr>
<tr>
<td>CSP-2010</td>
<td>3.1</td>
<td>1.5</td>
<td>1.0</td>
</tr>
<tr>
<td>MAXSAT12-PMS</td>
<td>8.6</td>
<td>6.5</td>
<td>2.7</td>
</tr>
<tr>
<td>PREMARSHALLING (OR)</td>
<td>2.3</td>
<td>2.9</td>
<td>3.6</td>
</tr>
<tr>
<td>PROTEUS-2014 (CSP)</td>
<td>6.5</td>
<td>10.9</td>
<td>6.3</td>
</tr>
</tbody>
</table>
Idea:

Find the best selector for given scenario,
Idea:

Find the best selector for given scenario, by automatically configuring highly parametric algorithm selector framework
**Idea:**

Find the best selector for given scenario, by automatically configuring highly parametric algorithm selector framework

[see also AutoWEKA – Thornton *et al.* 2013+17]
Algorithm configuration [see, e.g., Hutter et al. 2007]
Algorithm configuration [see, e.g., Hutter et al. 2007]

Instances $I$

Algorithm $A$, configuration space $C$

Select $c \in C$

Assess $A(c)$ on some $I' \subseteq I$

Performance

Best configuration $\hat{c}$

Configuration task
AutoFolio

Selection scenario with data $D$

Algorithm selector $AS$, configuration space $C$
AutoFolio

Selection scenario with data $D$

Split $D$ into folds $\{D_i\}_{i \in \{1...k\}}$

Algorithm selector $AS$, configuration space $C$
AutoFolio

Selection scenario with data $D$

Split $D$ into folds $\{D_i\}_{i \in \{1 \ldots k\}}$

Algorithm selector $AS$, configuration space $C$

Select $c \in C$, $i \in \{1 \ldots k\}$

Train $AS(c)$ on $D \setminus D_i$

Performance on $D_i$

Best configuration $\hat{c}$

Performance on $D_i$
Which choices?

AS approach

ML technique

Hyperparameters
Which choices?

AS approach

ML technique

Hyperparameters

Pre-solving

max. time

max. solvers

Lindauer, Hoos, Hutter, Schaub: AutoFolio: Algorithm Configuration for Algorithm Selection
Which choices?

- AS approach
- ML technique
- Hyperparameters
- Pre-solving
  - max. time
  - max. solvers
- Feature preproc.
Which choices?

- AS approach
- ML technique
- Hyperparameters

Pre-solving
- max. time
- max. solvers

Feature preproc.

Performance preproc.
Which choices?
Which choices?
Putting it all together

*AutoFolio* = configurator + selector framework
Putting it all together

**AutoFolio = configurator + selector framework**

Here:

- **SMAC** [Hutter et al. 2011] + **claspfolio 2** [Lindauer et al. 2014]
Putting it all together

AutoFolio = configurator + selector framework

Here:

$SMAC$ [Hutter et al. 2011] + $claspfolio 2$ [Lindauer et al. 2014]

Experimental evaluation:

- 13 ASlib scenarios
- 12 independent runs
- 2 days as configuration budget
- 10-fold outer cross validation
- 25 CPU years
Performance on SAT12-ALL

⇝ 2-fold speedup
Which Choices Lead To Good Performance?

fANOVA Analysis [Hutter et al. 2014] on SAT12-ALL:

1. max time for feature computation – 23.43% of variance
Which Choices Lead To Good Performance?

fANOVA Analysis [Hutter et al. 2014] on SAT12-ALL:

1. max time for feature computation – 23.43% of variance
2. algorithm filtering – 6.82% of variance
Which Choices Lead To Good Performance?

fANOVA Analysis [Hutter et al. 2014] on SAT12-ALL:

1. max time for feature computation – 23.43% of variance
2. algorithm filtering – 6.82% of variance
3. selection approach – 6.39% of variance
Which selection approaches?

- **CLUSTERING**
- **K-NN**
- **MULTI-CLASS CLASSIFICATION**
- **PAIRWISE CLASSIFICATION**
- **REGRESSION**

The chart shows the frequency of different selection approaches across various techniques:
- **SPECTRAL**
- **RANDOM FOREST**
- **SVM**
- **GRADIENT BOOSTING**

The bars represent the percentage frequency of each approach.
Observations

- similar results often achievable with lower effort:
  - 16 $\rightarrow$ 8 configurator runs
  - 48 $\rightarrow$ 24 hours configuration budget
  - SAT12-ALL: performance drop by only 8%
Observations

- similar results often achievable with lower effort:
  - 16 → 8 configurator runs
  - 48 → 24 hours configuration budget
  - SAT12-ALL: performance drop by only 8%

- depending on configuration budget, configure more or fewer parts
Observations

- similar results often achievable with lower effort:
  - 16 $\rightarrow$ 8 configurator runs
  - 48 $\rightarrow$ 24 hours configuration budget
  - $\Rightarrow$ SAT12-ALL: performance drop by only 8%

- depending on configuration budget, configure more or fewer parts

- robust performance:
  - established state-of-the-art performance on 7 scenarios
  - matches state-of-the-art performance on all other scenarios
AutoFolio 2.0

- https://github.com/mlindauer/AutoFolio
- open-source under BSD license
- easy to use; only two inputs
  - csv file for algorithm performance
  - csv file for instance features
AutoFolio 2.0

- [https://github.com/mlindauer/AutoFolio](https://github.com/mlindauer/AutoFolio)
- open-source under BSD license
- easy to use; only two inputs
  - csv file for algorithm performance
  - csv file for instance features
- reduced configuration space to most relevant parts
  ⇒ more efficient search
- reduced overhead by using SMAC3
AutoFolio 2.0

- https://github.com/mlindauer/AutoFolio
- open-source under BSD license
- easy to use; only two inputs
  - csv file for algorithm performance
  - csv file for instance features
- reduced configuration space to most relevant parts
  - more efficient search
- reduced overhead by using SMAC3
- accompanied by EDA tool to get insights into data:
  ASAPY: https://github.com/mlindauer/asapy
In a nutshell:

- different selectors are best for various selection scenarios
In a nutshell:

- different selectors are best for various selection scenarios
- automatically configure flexible selector framework to find good, custom selectors
In a nutshell:

- different selectors are best for various selection scenarios
- automatically configure flexible selector framework to find good, custom selectors
- good results using existing configurators, selector framework
- potential for further improvements
- meta-learning for warmstarting [Feurer et al. 2015]
- new ML approaches (e.g., XGBoost, DNNs)
- configure selectors for parallel portfolios [Lindauer et al. 2015]
In a nutshell:

- different selectors are best for various selection scenarios
- automatically configure flexible selector framework to find good, custom selectors
- good results using existing configurators, selector framework
- potential for further improvements
  - meta-learning for warmstarting [Feurer et al. 2015]
  - new ML approaches (e.g., XGBoost, DNNs)
  - configure selectors for parallel portfolios [Lindauer et al. 2015]