Sparkle: Accessible Meta-Algorithmics

Koen van der Blom\textsuperscript{1}, Chuan Luo\textsuperscript{2}, Holger H. Hoos\textsuperscript{1,3}  
2020-11-09

\textsuperscript{1}Leiden University  
\textsuperscript{2}Microsoft Research Asia  
\textsuperscript{3}University of British Columbia
Sparkle

• Meta-algorithmics for everyone
  • Algorithm selection
  • Algorithm configuration

• Benchmarking - Fair comparison of
  • Target algorithms (like CSSC) [Hutter et al. 2017]
  • Meta-algorithms (like AClib) [Hutter et al. 2014]

• Competitions (e.g. SAT, planning) [Luo et al. 2018, 2019]

• Best practice and avoid pitfalls [e.g. Eggensperger et al. 2019]
Algorithm configuration

- Get better performance
- Used incorrectly
  - “This doesn’t work!”
  - Wrong result (interpretation)
- Comparing algorithms
  - More can go wrong
- Comparing configurators
  - Even more can go wrong
AClib [Hutter et al 2014]

• Wrapper – Ensure algorithm calls are consistent across configurators

• Runsolver – Ensure runtime measurement is consistent

• Basic statistics and scatterplots (AClib 2)
Sparkle

• Report
  • Plots and statistics
  • What happens under the hood / practices used

• Integration with selection

• Just running target algorithms

• Analysis and validation tools
  • Integrated parameter importance analysis
  • Was the instance set homogeneous?
Sparkle

• Configuration and selection for everyone

• Benchmarking
  • Compare target- and meta-algorithms fairly

• Competitions (e.g. SAT, planning)

• Aid following best practice and avoiding pitfalls

• First release soon™: )