Accessible meta-algorithms with Sparkle

Koen van der Blom\textsuperscript{1,2}, Holger Hoos\textsuperscript{3,2,4}, Chuan Luo\textsuperscript{5}, Jeroen Rook\textsuperscript{6}

\textsuperscript{1}Sorbonne Université, \textsuperscript{2}Leiden University, \textsuperscript{3}RWTH Aachen University, \\
\textsuperscript{4}University of British Columbia, \textsuperscript{5}Beihang University, \textsuperscript{6}University of Twente
Meta-algorithms

• Algorithm configuration

• Algorithm selection

• Parallel algorithm portfolios

• Get the best performance out of algorithms
Adoption of meta-algorithms

• Adoption is limited, even in ML research [Bouthillier & Varoquaux, 2020]

• Meta-algorithms are complex and difficult for non-experts

• Substantial pitfalls, e.g., in AAC [Eggensperger et al., 2019]

• Errors are costly, e.g., re-running AAC is computationally expensive
Expanding the reach of meta-algorithms

• COSEAL community

• ...

• Laypeople from any other field

• Sparkle platform → Reach wider audiences, one step at a time
Goals of Sparkle

• Simplify the use of meta-algorithms
• Increase the adoption of meta-algorithms
• Prevent common pitfalls and often-made errors
• Ensure proper experimentation pipelines

• Improve our ability to assess, access and improve the SOTA in computational problem solving
Command line and scripting interface

1: Commands/initialise.py
2: Commands/add_instances.py path/to/PTN/
3: Commands/add_solver.py --deterministic 0 path/to/PbO-CSCCSAT/
4: Commands/add_solver.py --deterministic 0 path/to/MiniSAT/
5: Commands/add_feature_extractor.py path/to/Extractor/
6: Commands/compute_features.py
7: Commands/run_solvers.py
8: Commands/construct_sparkle_portfolio_selector.py
9: Commands/generate_report.py
Reports

- Solver, instance set(s)
- Configurator, protocols, budgets
- Final configuration
- Performance measures
- Comparison plots
- References
- # timeouts
- Ablation analysis (optional)
Going forward

- Slurm clusters only → Local execution in progress
- SMAC only → Extending to other configurators
- Simplify, improve, extend ... in many directions
Sparkle makes meta-algorithms accessible for improving the state of the art in solving challenging problems in AI.

Try Sparkle yourself!

bitbucket.org/sparkle-ai/sparkle

Opening for research engineer @ RWTH Aachen