Accessible and Robust Multi-Objective Automated Algorithm Selection and Configuration

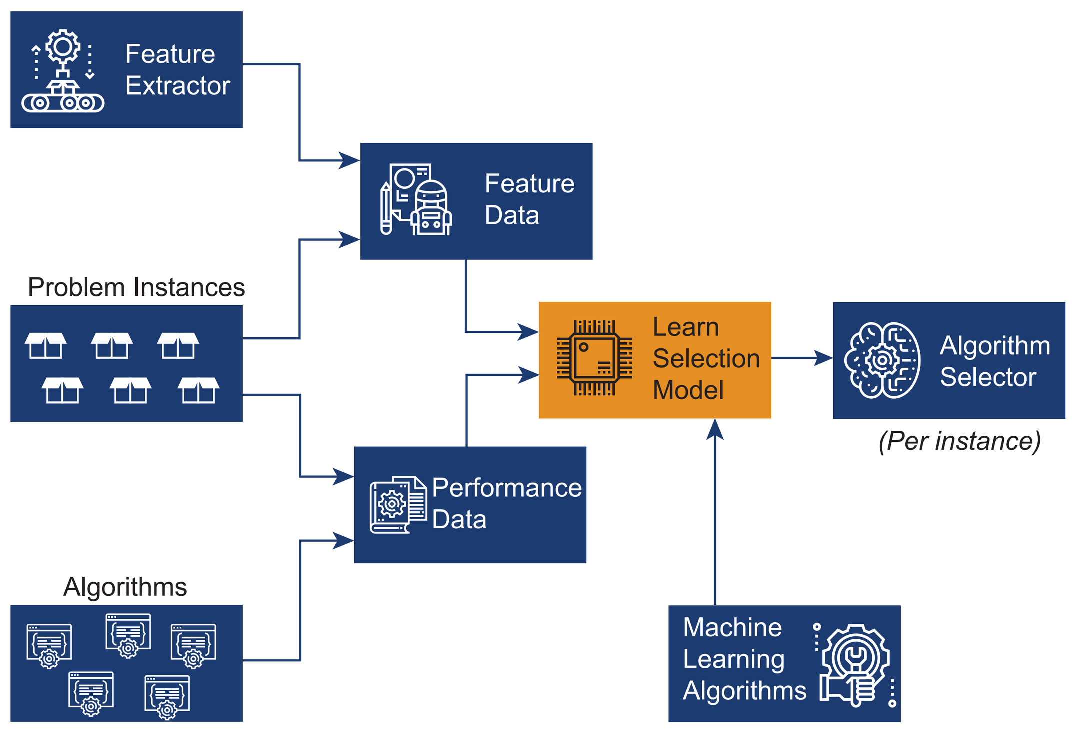
Introduction

- Most algorithms have parameters that affect the performance and manually tuning them is a tedious endeavour.
- Comparing algorithms is often based on their ranking, based on the aggregated performance score on a benchmark set of problem instances.

Meta-algorithmic techniques, such as automated algorithm configuration and selection enable performance optimisation and algorithm comparison in a **sophisticated, insightful and robust** manner. These methods are complex to use, which lacks widespread adoption. We present Sparkle: a Programming by Optimisation platform that enables easy access to meta-algorithmics.

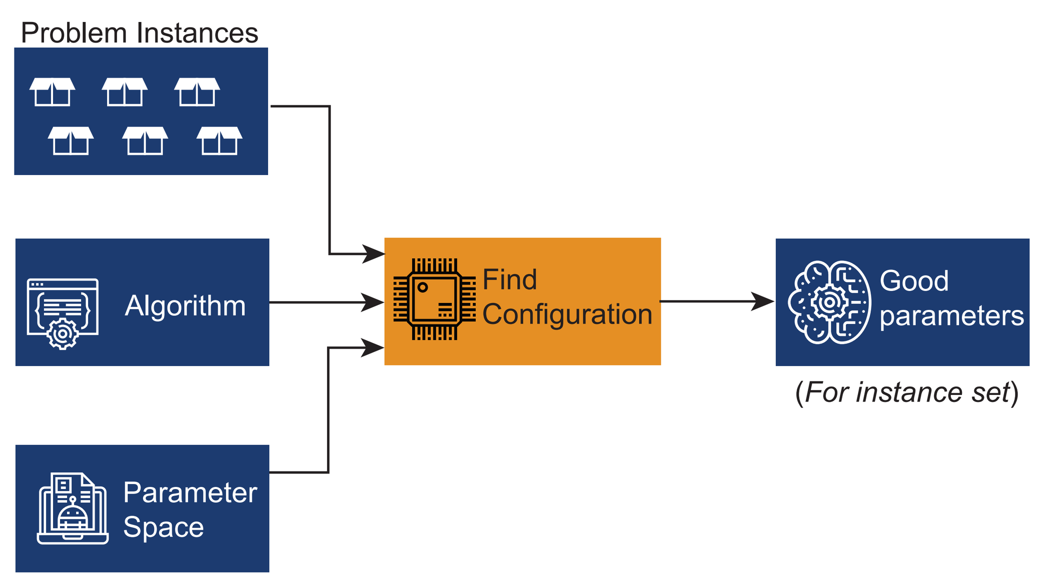
Automated Algorithm Selection

A single algorithm is rarely always better than all other algorithms. Hence, it is beneficial to have a portfolio of (optimization) algorithms that are complementary w.r.t. their performance. **Automated Algorithm Selection (AAS) combines algorithms into a portfolio** and for each problem instance predicts which one yields the best performance**.**



Automated Algorithm Configuration

Algorithms have performance affecting parameters. The number of possible parameter configurations is often very large. **Automated Algorithm Configuration (AAC) efficiently finds a good set of parameters** from the parameter space that maximises the performance for a class of problems and is visualised below:



Sparkle Platform

Sparkle is a framework that eases the use of AAS and AAC techniques by **standardizing** and **automating** the experimental set-up. This ensures the proper use of these methods and prevents, probably costly, human errors.

Combining both AAS and AAC into a single framework enables further improvement regarding the effective combination of AAS and AAC techniques. Additionally, a wide range of other uses is included, such as **benchmarking algorithms** based on their **marginal contribution** in a per-instance algorithm selector.

Interaction with the platform is done via simple **self-explanatory commands**, which makes experiments conducted with Sparkle scriptable.

```
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/configure_solver.py --solver
   PbO-CSCCSAT --instance-set-train PTN
5. Commands/generate report.py
```

Sparkle compiles the experimantal setup and presents the results into a nicely formated report. Additional experiments, such as determining parameter importance are added to the report as well if they were conducted.

Sparkle is available at:
bitbucket.org/sparkle-ai/sparkle/

Try out Sparkle:

Projects

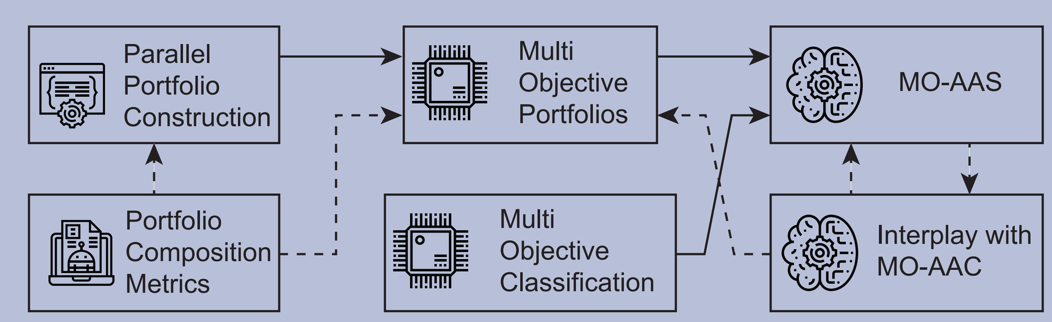
With multiple projects we are adding new functionalities to the Sparkle platform. Our focus is on adding support for multi-objective optimization algorithms, multi-objective performance measurement, robust algorithm comparison and configuration.

Robust Algorithm Comparison

Comparing algorithms is challenging. A typical algorithm comparison consists of running them on multiple problem instances and then aggregating the results into a single score. Algorithms are then ranked based on that score. The choice of instances is highly influential on the outcome of this ranking, making such approaches statistically weak. We plan to obtain statistically robust comparions based on **bootstrap distributions**, by considering **stochastic algorithm behaviour** and by ranking based on **multiple performance objectives**.

Multi-Objective AAC and AAS

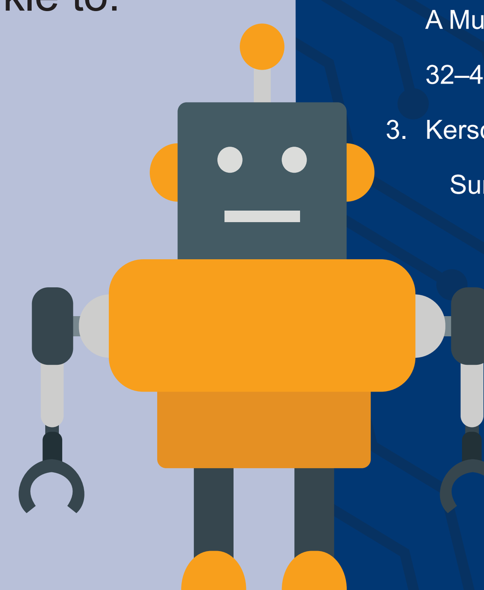
There are often many different measures for evaluating the performance of an algorithm. These performance metrics are dependent on the problem domain. Some examples are; **running time, solution quality, and solving probability**. Often, there are **trade-offs** between these metrics, making it a multi-objective optimisation problem.
In the diagram below the path to obtain such methods is drawn:



Applications

To challenge the capabilities of Sparkle and to further improve on existing meta-algorithmic methods for specific problem domains we also apply Sparkle to:

- **Multi-objective Multimodal Optimisation**
- **Sparse Neural Networks**
- **TSP**
- **SAT**



Prof. dr. Heike Trautmann

Contact

h.trautmann@utwente.nl
people.utwente.nl/h.trautmann

Jeroen Rook MSc.

Contact

j.g.rook@utwente.nl
people.utwente.nl/j.g.rook

Sparkle is created in cooperation with
Prof. dr. Holger H. Hoos, RWTH Aachen & Leiden University
Dr. Koen van der Blom, Leiden University
Dr. Chuan Luo, Beihang University

To make the adoption of **automated algorithm configuration and selection** more widespread we created the **Sparkle platform.**
By combining meta-algorithmic approaches and by adding **multi-objective** support we make these methods more **robust** and **versatile.**

References

1. Rook, J.G., Trautmann, H., Bossek J., & Grimme C. (2022). On the Potential of Automated Algorithm Configuration on Multi-Modal Multi-Objective Optimization Problems. GECCO '22, [to appear in companion proceedings].
2. Blot, A., Hoos ,H. H., Jourdan, L., Kessaci-Marmion, M., & Trautmann, H. (2016). MO-ParamILS: A Multi-objective Automatic Algorithm Configuration Framework. LION 10. LNCS, Volume 10079, 32–47.
3. Kerschke, P., Hoos, H. H., Neumann, F., & Trautmann, H. (2019). Automated Algorithm Selection: Survey and Perspectives. Evolutionary Computation (ECJ), 27(1), 3–45.
4. Kerschke, P., Kotthoff, L., Bossek, J., Hoos, H. H., & Trautmann, H. (2018). Leveraging TSP Solver Complementarity through Machine Learning. Evolutionary Computation (ECJ), 26(4), 597–620.
5. Bossek, J., Kerschke, P., & Trautmann, H. (2020). A Multi-Objective Perspective on Performance Assessment and Automated Selection of Single-Objective Optimization Algorithms. Applied Soft Computing, 2020(88), 105901.