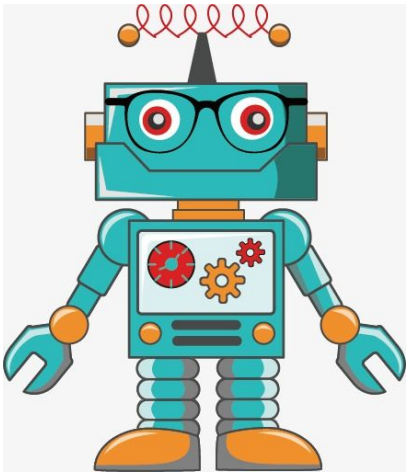
Automating SAT Solver Research

markus.iser@kit.edu jakob.bach@kit.edu

An infinite amount of gorillas produces infinite amounts of code. Eventually, some of this code compiles into programs, and some of these programs are SAT solvers, which means that they correctly solve the NP-complete SAT problem.

Solbert the gorilla writing SAT Solvers

A research robot incrementally evaluates the efficiency of the SAT solvers which are produced by the gorillas. It deletes those solvers which are not efficient enough and combines the most efficient solvers to a portfolio of solvers [1]. As new solvers emerge, the robot repeats this process and thus continuously improves its solver portfolio.



Research Robot

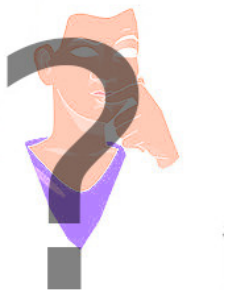
Whenever a gorilla creates a new SAT solver, the research robot determines the efficiency of the solver by measuring its runtimes on a large set I of benchmark instances [2]. The runtime per instance is limited by T, such that the robot records either a valid runtime or T.

The robot maintains a database of runtimes $r : I \times S \to [0, T]$ of solvers S. In addition to the runtimes r, the robot maintains a set of instance feature records $f : I \to D^n$. The robot uses f and r to organize the benchmark instances into clusters of similar instances which are solved by mostly the same solvers. For each such cluster of instances, the robot maintains a list of solvers which are best for solving instances in that cluster.

With an increasing number of benchmark instances, solvers, and solver configurations, the number of required runtime experiments is subject to super-linear growth. Moreover, timeouts of one hour and more are very common and the research robot has only a limited number of computers. Last but not least, resource usage comes with a cost: If the research robot gets into debt, the climate changes and all the gorillas die.

As a human researcher, you can help us to improve the research robot. While conducting a *minimal amount of solver runtime experiments*, the robot should make *outcome predictions of high confidence*. For each cluster, you analyze the minimal number of experiments for outcome prediction of bounded error for several bounds. Additionally, you conduct case-studies for variants of the given clustering and prediction methods.

You will work in an interdisciplinary environment between research on deductive AI (SAT Algorithms) and inductive AI (Big Data Analysis). You get the chance to use state-of-the-art prediction models, clustering techniques and feature selection methods. Help us to build a cost-efficient research robot and save the gorillas!



Wanted: Human Researcher

References

[1] Collautti, M., Malitsky, Y., Mehta, D., O'Sullivan, B.: SNNAP: solver-based nearest neighbor for algorithm portfolios. In: Blockeel, H., Kersting, K., Nijssen, S., Železný, F. (eds.) Machine Learning and Knowledge Discovery in Databases - European Conference, ECML PKDD 2013, Prague, Czech Republic, September 23-27, 2013, Proceedings, Part III. Lecture Notes in Computer Science, vol. 8190, pp. 435–450. Springer (2013)

[2] Iser, M., Springer, L., Sinz, C.: Collaborative management of benchmark instances and their attributes. CoRR **abs/2009.02995** (2020), https://arxiv.org/abs/2009.02995