

Configuration Report for the Solver PbO-CCSAT-Generic on the Training Instance Set PTN and Testing Instance Set PTN2 in *Sparkle*

Automatically generated by *Sparkle* (version: 0.8.1)

17th June 2024

1 Introduction

Sparkle [2] is a multi-agent problem-solving platform based on Programming by Optimisation (PbO) [1], and would provide a number of effective algorithm optimisation techniques (such as automated algorithm configuration, portfolio-based algorithm selection, etc) to accelerate the existing solvers.

This experimental report is automatically generated by *Sparkle*. This report presents experimental results on the scenario of configuring the solver PbO-CCSAT-Generic on the training instance set PTN and evaluating it on the testing instance set PTN2.

2 Information about the Instance Set(s)

- Training set: **PTN**, consisting of 12 instances
- Testing set: **PTN2**, consisting of 11 instances

3 Information about the Configuration Protocol

The configurator used in *Sparkle* is SMAC (*Sequential Model-based Algorithm Configuration*) [3], and the version of SMAC used in *Sparkle* is 2.10.03.

During the configuration process, *Sparkle* performs 25 independent SMAC runs for configuring the solver PbO-CCSAT-Generic on the training instance set PTN; the configuration objective is RUN-TIME; the whole configuration time budget is 600 seconds; the cutoff time for each run is 60 seconds.

Each independent run of SMAC would result in one optimised configuration. As a result, *Sparkle* would obtain 25 optimised configurations. Each of these was then evaluated on the entire training set, with one solver run per instance and a cutoff time of 60 seconds, and the configuration with the lowest PAR10 value was selected as the result of the configuration process.

4 Information about the Optimised Configuration

After the configuration process mentioned above, *Sparkle* obtained the optimised configuration. The details of the optimised configuration are described below.

```
-p_swt '0.37822764235422035' -perform_aspiration '1' -perform_clause_weight '1' -perform_double_cc
'1' -perform_first_div '0' -perform_pac '0' -q_swt '0.5433225022192605' -sel_clause_div '2' -sel_clause_weight_scheme
'1' -sel_var_break_tie_greedy '1' -sel_var_div '2' -threshold_swt '592'
```

5 Comparison between Configured Version and Default Version on the Training Instance Set

In order to investigate the performance on the training instance set, *Sparkle* ran the configured version of PbO-CCSAT-Generic and the default version of PbO-CCSAT-Generic on the training instance set. During this phase, each version performed one run per instance with a cutoff time of 60 seconds. The results are reported as follows.

- **PbO-CCSAT-Generic (configured)**, PAR10: 55.45833300000001
- **PbO-CCSAT-Generic (default)**, PAR10: 226.99671249999997

The empirical comparison between the PbO-CCSAT-Generic (configured) and PbO-CCSAT-Generic (default) on the training set of PTN is presented in Figure 1.

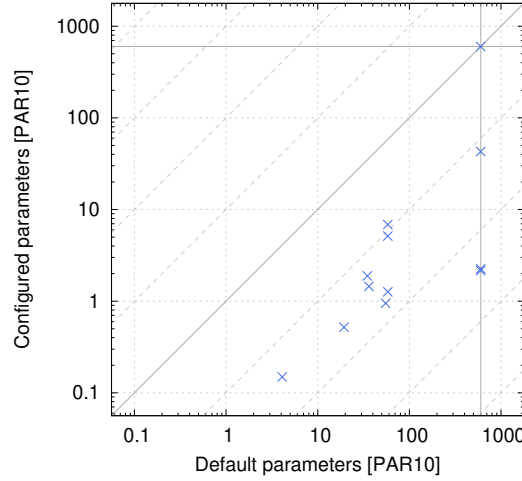


Figure 1: Empirical comparison between the PbO-CCSAT-Generic (configured) and PbO-CCSAT-Generic (default) on the training set of PTN.

Table 1 shows on how many instances the PbO-CCSAT-Generic (configured) and PbO-CCSAT-Generic (default) timed out (did not solve the instance within the cutoff time of 60 seconds) on the training set of PTN, as well as on how many instances both timed out.

configured	default	overlap
1	4	1

Table 1: Number of time-outs for PbO-CCSAT-Generic (configured), PbO-CCSAT-Generic (default), and for how many instances both timed out on the training set of PTN.

6 Comparison between Configured Version and Default Version on the Testing Instance Set

After specifying the optimised configuration, *Sparkle* ran the configured version of PbO-CCSAT-Generic and the default version of PbO-CCSAT-Generic on the testing instance set. During this phase, each version performed one run per instance with a cutoff time of 60 seconds. The results are reported as follows.

- **PbO-CCSAT-Generic (configured)**, PAR10: 9.619044545454546
- **PbO-CCSAT-Generic (default)**, PAR10: 130.26663809090908

The empirical comparison between the PbO-CCSAT-Generic (configured) and PbO-CCSAT-Generic (default) on the testing set of PTN2 is presented in Figure 2.

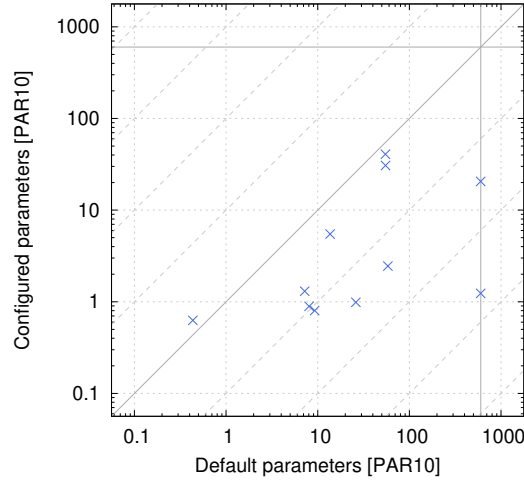


Figure 2: Empirical comparison between the PbO-CCSAT-Generic (configured) and PbO-CCSAT-Generic (default) on the testing set of PTN2.

Table 2 shows on how many instances the PbO-CCSAT-Generic (configured) and PbO-CCSAT-Generic (default) timed out (did not solve the instance within the cutoff time of 60 seconds) on the testing set of PTN2, as well as on how many instances both timed out.

configured	default	overlap
0	2	0

Table 2: Number of time-outs for PbO-CCSAT-Generic (configured), PbO-CCSAT-Generic (default), and for how many instances both timed out on the testing set of PTN2.

References

- [1] Holger H. Hoos. Programming by Optimization. *Communications of the ACM*, 55(2):70–80, 2012.

- [2] Holger H. Hoos. Sparkle: A pbo-based multi-agent problem-solving platform. Technical report, Department of Computer Science, University of British Columbia, 2015.
- [3] Frank Hutter, Holger H. Hoos, and Kevin Leyton-Brown. Sequential model-based optimization for general algorithm configuration. In *Proceedings of the 5th International Conference on Learning and Intelligent Optimization (LION 5)*, pages 507–523, 2011.