Experimental Reports for *Sparkle*

Sparkle

14th February 2018

1 Introduction

Sparkle [3] is a Programming by Optimisation (PbO) [2] multi-agent problem-solving platform, and would provide many algorithm optimisation techniques (such as automated algorithm configuration, portfolio-based algorithm selection, etc) to light your solvers.

This is the automatically generated report by *Sparkle*, and this report is to present experimental results for solvers and instance classes submitted to *Sparkle*.

2 Experimental Preliminaries

In this section, we present the experimental preliminaries, including the list of solvers, the list of feature extractors, the list of instance classes, information about experimental setup and the information about how to construct a portfolio-based algorithm selector in *Sparkle*.

2.1 Solvers

There are 4 solver(s) submitted in *Sparkle*, and the list of solvers are given as follows.

1. **PbO-CCSAT-FAC**
2. **PbO-CCSAT-SMT-QF-BV**
3. **Lingeling_wrapper_sparkle**
4. **PbO-CCSAT-PTN**

2.2 Feature Extractors

There are 1 feature extractor(s) submitted in *Sparkle*, and list of feature extractors are given as follow.

1. **SAT-features-competition2012_sparkle**

2.3 Instance Classes

There are 1 instance class(es) submitted in *Sparkle*, and the list of instance classes are given as follows.

1. **PTN-7824_Train**, number of instances: 11

2.4 Experimental Setup

Feature computation: We use all the feature extractors which are presented above to compute the feature vector for each instance. Each feature extractor will compute a feature vector for each instance. The final feature vector is the combination of all computed feature vectors. The cutoff time for feature vector computation on each instance is set to 90 seconds.

Performance computation: Each solver will run one time on each instance. The cutoff time for each performance computation run is set to 50 seconds.

2.5 Constructing Portfolio-Based Algorithm Selector

Sparkle runs all the feature extractors to compute the feature vector for each instance, and store the resulting the feature data in the system. Also, *Sparkle* runs all the solvers to solve each instance, and store the resulting the performance data in the system. After the feature-related and the performance-related experiments done, by utilizing the computed feature data and performance data, *Sparkle* uses *AutoFolio* [4] to automatically construct a portfolio-based algorithm selector for *Sparkle*.

3 Experimental Results

In this section, the related experimental results in *Sparkle* are presented and analysed.

3.1 PAR10 Ranking List

The ranking list with regards to the penalised average runtime (PAR10) for solvers is given as follows.

1. **Solvers/PbO-CCSAT-PTN**, PAR10: 3.06272727273
2. **Solvers/PbO-CCSAT-FAC**, PAR10: 93.27
3. **Solvers/PbO-CCSAT-SMT-QF-BV**, PAR10: 139.378181818
4. **Solvers/Lingeling_wrapper_sparkle**, PAR10: 500.0

Also, PAR10 for the Virtual Best Solver *VBS*, i.e., the perfect portfolio selector, and the actual portfolio selector in *Sparkle* is given as follows.

- ***VBS***, PAR10: 3.04727272727
- **Actual Portfolio Selector in *Sparkle***, PAR10: 3.06272727273

3.2 Marginal Contribution Ranking List

Sparkle uses the concept of marginal contribution [5] to measure each solver's contribution to the ***VBS*** and to the **actual portfolio selector in *Sparkle***. In this report, we uses the approach described in the literature [1] to each solver's marginal contribution.

Solver ranking list via marginal contribution [5] for solvers with regards to the ***VBS*** is given as follows.

1. **PbO-CCSAT-PTN**, marginal contribution: 2.03823716493
2. **PbO-CCSAT-SMT-QF-BV**, marginal contribution: 7.72376192639e-05
3. **PbO-CCSAT-FAC**, marginal contribution: 0.0

4. **Lingeling_wrapper_sparkle**, marginal contribution: 0.0

Solver ranking list via marginal contribution [5] for solvers with regards to the **actual portfolio selector in *Sparkle*** is given as follows.

1. **PbO-CCSAT-PTN**, marginal contribution: 2.04192639709
2. **PbO-CCSAT-FAC**, marginal contribution: 0.0
3. **PbO-CCSAT-SMT-QF-BV**, marginal contribution: 0.0
4. **Lingeling_wrapper_sparkle**, marginal contribution: 0.0

3.3 Scatter Plot Analysis

The empirical comparison between the actual portfolio selector in *Sparkle* and single best solver (*SBS*) is presented in Figure 1 The empirical comparison between the actual portfolio selector in *Sparkle* and *VBS* is presented in Figure 2.

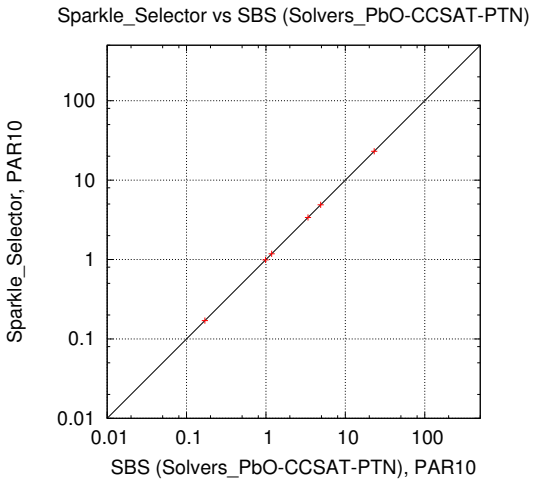

Figure 1: Empirical comparison between the actual portfolio selector in *Sparkle* and *SBS*.

References

[1] Alexandre Fréchette, Lars Kotthoff, Tomasz P. Michalak, Talal Rahwan, Holger H. Hoos, and Kevin Leyton-Brown. Using the shapley value to analyze algorithm portfolios. In *Proceedings of the 30th AAAI Conference on Artificial Intelligence (AAAI-16)*, pages 3397–3403. AAAI Press, 2016.

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

[3] Holger H. Hoos. Sparkle: A pbo-based multi-agent problem-solving platform. Technical report, Department of Computer Science, University of British Columbia, 2015.

[4] Marius Thomas Lindauer, Holger H. Hoos, Frank Hutter, and Torsten Schaub. AutoFolio: An automatically configured algorithm selector. *Journal of Artificial Intelligence Research*, 53:745–778, 2015.

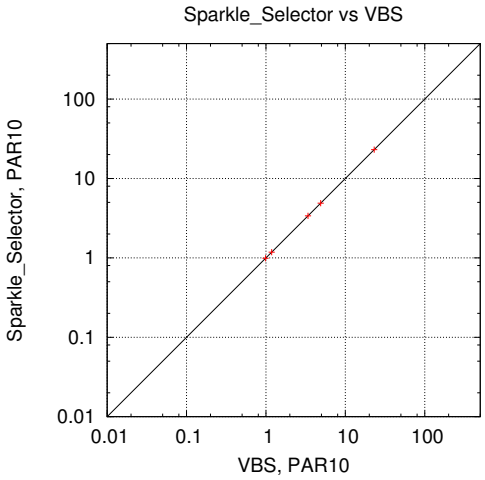

Figure 2: Empirical comparison between the actual portfolio selector in *Sparkle* and *VBS*.

[5] Lin Xu, Frank Hutter, Holger Hoos, and Kevin Leyton-Brown. Evaluating component solver contributions to portfolio-based algorithm selectors. In Alessandro Cimatti and Roberto Sebastiani, editors, *Theory and Applications of Satisfiability Testing - SAT 2012 - 15th International Conference, Trento, Italy, June 17-20, 2012. Proceedings*, volume 7317 of *Lecture Notes in Computer Science*, pages 228–241. Springer, 2012.