Configuration Space, Default Configuration and Optimised Configurations of *PbO-CCSAT*

Chuan Luo[1,2], Holger H. Hoos[2], and Shaowei Cai[3]

[1] Microsoft Research, China
[2] Leiden Institute of Advanced Computer Science, Leiden University, The Netherlands
[3] State Key Laboratory of Computer Science, Institute of Software, Chinese Academy of Sciences, China
chuan.luo@microsoft.com, hh@liacs.nl, caisw@ios.ac.cn

1 Configuration Space and Default Configuration

PbO-CCSAT can be seen as a CC-based meta solver that can be configured to instantiate various high-performance CC-based SAT solvers. We have introduced all components of our *PbO-CCSAT* framework in the paper. In Table 1, we give an overview of the full configuration space of *PbO-CCSAT* (*i.e.,* all heuristics and parameters, as well as the conditions when heuristics and parameters are activated). The table also shows the defaults for all parameters and design choices; we chose these to instantiate a version of *DCCASat* that is known to perform well on structured SAT instances [2], in order to provide a strong starting point for our automated configuration process (described in detail in the following section). More particularly, we list the configuration settings of *PbO-CCSAT (Default)* (*i.e., DCCASat* for structured SAT instances) in Table 2.

2 Optimised Configurations for All Testing Benchmarks

PbO-CCSAT has been developed as a highly parametric SLS framework for SAT, in order to be automatically configured to perform well on different classes of SAT instances. Towards this end, we made use of *SMAC* (version: 2.10.03), a state-of-the-art, general-purpose algorithm configurator based on the concept of sequential model-based optimisation (also known as Bayesian optimisation) [1]. We now describe the protocol we used for *PbO-CCSAT* and all other solvers described in the paper.

Following standard practice, and to achieve a balance between the number of instances solved within a given cutoff time and the actual running time required for solving them, we used SMAC to minimise PAR10 (*i.e.,*, average running time, where unsuccessful runs are counted at 10 times the cutoff time). We used an overall time budget of 36 000 seconds (= 10 hours) for each run of SMAC, and a cutoff of 60 seconds per solver run during configuration. For each training set, we performed 25 independent runs of *SMAC*, resulting in 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 CPU seconds per run, and the configuration with the lowest PAR10 value was selected as the result of the

Table 1: The configuration space of *PbO-CCSAT*. Each row corresponds to a design choice controlled by a parameters; P denotes numerical and Boolean parameters, while H denotes heuristics whose instantiation is determined by a categorical parameter.

Heuristic/Parameter	Activated Conditions	Type	Domain	Default Value
performRW (P)	None	Boolean	{*True*,*False*}	*False*
performProbDiv (P)	None	Boolean	{*True*,*False*}	*False*
performCSCC (P)	None	Boolean	{*True*,*False*}	*True*
performAspiration (P)	None	Boolean	{*True*,*False*}	*True*
performCWS (P)	None	Boolean	{*True*,*False*}	*True*
asgnGenHeur (H)	None	Categorical	{1,2}	1
selUnsatClause (H)	None	Categorical	{1,2}	1
selVarFromUnsatClause (H)	None	Categorical	{1,2,3,4,5,6,7}	3
selVarFromSet (H)	None	Categorical	{1,2,3,4}	2
clauseWeightScheme (H)	*performCWS* = *True*	Categorical	{1,2}	1
rw_prob (P)	*performRW* = *True*	Real	[0.00001, 0.1]	0.00058
div_prob (P)	*performProbDiv* = *True*	Real	[0.001, 1]	0.0228
hscore_d (P)	*selVarFromSet*=3 or *selVarFromUnsatClause*=4	Integer	[1,15]	8
$hscore\_\beta$ (P)	*selVarFromSet*=3 or *selVarFromUnsatClause*=4	Integer	[100,1 000 000]	1 000
$hscore_2\_\gamma$ (P)	*selVarFromSet*=4 or *selVarFromUnsatClause*=5	Integer	[100,1 000 000]	1 000
swt_threshold (P)	*performCWS* = *True* and *clauseWeightScheme*=1	Integer	[10,1 000]	300
swt_p (P)	*performCWS* = *True* and *clauseWeightScheme*=1	Real	[0,1]	0.3
swt_q (P)	*performCWS* = *True* and *clauseWeightScheme*=1	Real	[0,1]	0
paws_sp (P)	*performCWS* = *True* and *clauseWeightScheme*=2	Real	[0.5,1]	0.8
novelty_prob (P)	*selVarFromUnsatClause*=6	Real	[0,1]	[0.119]
sp_c_1 (P)	*selVarFromUnsatClause*=7	Real	[2,10]	[2.15]
sp_c_2 (P)	*selVarFromUnsatClause*=7	Integer	[1,5]	[4]
sp_c_3 (P)	*selVarFromUnsatClause*=7	Integer	[20 000,100 000]	[75 000]

Table 2: The configuration settings of *PbO-CCSAT (Default)* (*i.e., DCCASat* for structured SAT instances).

Instantiation	Configuration Settings
PbO-CCSAT (Default)	*performRW* = *False*, *performProbDiv* = *False*, *performCSCC* = *True*, *performAspiration* = *True*, *performCWS* = *True*, *asgnGenHeur*=1, *selUnsatClause*=1, *selVarFromUnsatClause*=3, *selVarFromSet*=2, *clauseWeightScheme*=1, *swt_threshold*=300, *swt_p*=0.3, *swt_q*=0

configuration process. As per SMAC's default settings, we performed one run per problem instance during this validation phase when configuring algorithms. The configurations of *PbO-CCSAT* obtained in this way for all our training benchmarks are reported in Table 3.

References

1. Hutter, F., Hoos, H.H., Leyton-Brown, K.: Sequential model-based optimization for general algorithm configuration. In: Proceedings of LION 2011. pp. 507–523 (2011)
2. Luo, C., Cai, S., Wu, W., Su, K.: Double configuration checking in stochastic local search for satisfiability. In: Proceedings of AAAI 2014. pp. 2703–2709 (2014)

Table 3: The optimised configuration settings of *PbO-CCSAT* for all training benchmarks found by *SMAC*.

Training Benchmark	Tuned Configuration Settings
FCC-SAT [Train]	*performRW* = *False*, *performProbDiv* = *False*, *performCSCC* = *False*, *performAspiration* = *True*, *performCWS* = *True*, *asgnGenHeur*=1, *selUnsatClause*=2, *selVarFromUnsatClause*=6, *selVarFromSet*=1, *clauseWeightScheme*=1, *swt_threshold*=450, *swt_p*=0.09697519163726609, *swt_q*=0.2009167390247052, *novelty_prob*=0.12032578283349094
PTN [Train]	*performRW* = *False*, *performProbDiv* = *True*, *performCSCC* = *False*, *performAspiration* = *False*, *performCWS* = *True*, *asgnGenHeur*=1, *selUnsatClause*=1, *selVarFromUnsatClause*=1, *selVarFromSet*=1, *clauseWeightScheme*=1, *div_prob*=0.04267006927278742, *swt_threshold*=52, *swt_p*=0.10305467099383325, *swt_q*=0.7462070417690693
SMT-QF-BV [Train]	*performRW* = *False*, *performProbDiv* = *True*, *performCSCC* = *True*, *performAspiration* = *True*, *performCWS* = *True*, *asgnGenHeur*=2, *selUnsatClause*=1, *selVarFromUnsatClause*=1, *selVarFromSet*=1, *clauseWeightScheme*=1, *div_prob*=0.05615250814954268, *swt_threshold*=958, *swt_p*=0.1629585205916365, *swt_q*=0.08020153811108388
Community [Train]	*performRW* = *True*, *performProbDiv* = *True*, *performCSCC* = *False*, *performAspiration* = *True*, *performCWS* = *True*, *asgnGenHeur*=1, *selUnsatClause*=2, *selVarFromUnsatClause*=1, *selVarFromSet*=4, *clauseWeightScheme*=1, *rw_prob*=1.5063535410167932e-5, *div_prob*=0.003546954428490734, *swt_threshold*=613, *swt_p*=0.5835271379082203, *swt_q*=0.29666262025509293, $hscore_2\_\gamma$=539 093
SC17-Main-mp1-9 [Train]	*performRW* = *False*, *performProbDiv* = *False*, *performCSCC* = *False*, *performAspiration* = *False*, *performCWS* = *True*, *asgnGenHeur*=2, *selUnsatClause*=1, *selVarFromUnsatClause*=3, *selVarFromSet*=1, *clauseWeightScheme*=2, *paws_sp*=0.6877076185004205