SAT-Race 2010 Solver Description: `borg-sat-10.06.07`

Bryan Silverthorn and **Risto Miikkulainen**
Department of Computer Science
The University of Texas at Austin
{bsilvert,risto}@cs.utexas.edu

Introduction

Algorithm portfolio methods (Huberman, Lukose, and Hogg 1997) use information about solvers and problem instances to allocate computational resources among multiple solvers, attempting to maximize the time spent on those well suited to each instance. Portfolio methods such as SATzilla (Xu et al. 2008) have proved increasingly effective in satisfiability.

An algorithm portfolio must decide which solvers to run and for how long to run them. These decisions rely entirely on expectations about solver behavior.

The `borg-sat` solver attempts to to learn predictable aspects of solver behavior—such as how likely a solver is to succeed if it has previously failed—given data on the successes and failures of solvers on many problem instances. The version of this solver submitted to SAT-Race 2010, `borg-sat-10.06.07`, assumes a specific *latent class* model of solver behavior, a mixture of Dirichlet compound multinomial (DCM) distributions, which is used to identify groups of similar problem instances. This model is examined in detail by Silverthorn and Miikkulainen (2010). It captures the basic correlations between solvers, runs, and problem instances, as well as the tendency of solver outcomes to recur. Unlike the classifier employed by SATzilla, the model considers only the success or failure of each past solver run; it does *not* consider instance feature information.

This version of `borg-sat` employs the DCM mixture model in computing an optimal fixed-length solver execution schedule followed for every problem instance, as described in the following section.

Computing an Execution Schedule

Predictions of solver performance are useful only if they can be used to execute more appropriate solvers more often. To describe the algorithm portfolio situation in decision-theoretic terms, we take our set of past observations—in this case, the solvers already executed on this problem instance, and their success or failure—as our *belief state*. The Bellman equation describes the expected reward of an optimal policy,

$$V^*(s) = R(s) + \max_a \gamma \sum_{s'} P(s'|s,a)V^*(s'),$$

where s is a particular belief state, $R(s)$ describes the reward associated with a state (here, let 1 be the reward of any state in which any solver has been successful, and 0 the reward otherwise), γ is an arbitrary discount factor (which can be set lower to prefer quickly-obtained solutions more strongly), a is an action (the execution of some solver for some amount of time), and $P(s'|s,a)$ is the probability of arriving in state s' after taking action a in state s. Since the number of possible belief states grows quickly as actions are taken, the optimal policy is practical to compute only if the portfolio is limited to a short action sequence.

Just such an optimal short sequence of actions was computed offline, using a learned DCM model to define P; the `borg-sat-10.06.07` solver then follows that sequence when solving any new problem instance.

Portfolio Composition

Portfolio methods rely entirely on the performance of the solvers they employ, and are possible only because of the engineering and research involved in making those solvers effective. This version of `borg-sat` considered 13 subsolvers in its model: every qualifying solver in the application category of the final round of the 2009 SAT competition, excluding the reference solvers and SATzilla, with two exceptions (`kw` and `MiniSat 2.1`), as well as two more recent solvers (`cryptominisat-2.4.2` and `precosat-465r2-2ce82ba-100514`). Table 1 lists these solvers and their authors. The final policy did not use every solver, and the `SatELite` preprocessor was applied before solver execution.

Acknowledgments

This research was supported in part by the NSF under grants EIA-0303609, IIS-0757479, and IIS-0915038, and by the THECB under grant 003658-0036-2007.

Details

References

Huberman, B.; Lukose, R.; and Hogg, T. 1997. Economics Approach to Hard Computational Problems. *Science*.

Silverthorn, B., and Miikkulainen, R. 2010. Latent Class Models for Algorithm Portfolio Methods. In *AAAI*.

Xu, L.; Hutter, F.; Hoos, H. H.; and Leyton-Brown, K. 2008. SATzilla: Portfolio-based Algorithm Selection for SAT. *JAIR*.

Name	Reference
`CircUs 2009-03-23`	Hyojung Han
`clasp 1.2.0-SAT09-32`	Benjamin Kaufmann
`glucose 1.0`	Gilles Audemard and Laurent Simon
`LySAT i/2009-03-20`	Youssef Hamadi, Saïd Jabbour, and Lakhdar Saïs
`ManySAT 1.1 aimd 1/2009-03-20`	Youssef Hamadi, Saïd Jabbour, and Lakhdar Saïs
`MiniSAT 09z 2009-03-22`	Markus Iser
`minisat_cumr p-2009-03-18`	Kazuya Masuda and Tomio Kamada
`MXC 2009-03-10`	David Bregman
`precosat 236`	Armin Biere
`Rsat 2009-03-22`	Knot Pipatsrisawat and Adnan Darwiche
`SApperloT base`	Stephan Kottler
`cryptominisat-2.4.2`	Mate Soos
`precosat-465r2-2ce82ba-100514`	Armin Biere

Table 1: Subsolvers considered by the `borg-sat-10.06.07` planner.