Dynamic Scoring Functions with Variable Expressions: New SLS Methods for Solving SAT (ONLINE APPENDIX)

Dave A. D. Tompkins and Holger H. Hoos
Department of Computer Science
University of British Columbia
{davet,hoos}@cs.ubc.ca

June 17, 2010

Version 0.1 (preliminary) – This will be updated to be consistent with Dave's PhD dissertation once finalized. As it is currently, it is (mostly) cut & pasted from his external examiner draft.

1 UBC Arrow Cluster

The UBC arrow cluster at UBC is composed of 55 dual 3.2GHz Intel Xeon PCs with 2GB RAM, 2MB cache, running SuSE Linux.

2 WestGrid glacier cluster

The WestGrid glacier cluster is composed of 840 computational nodes, each with two 3.06 GHz Intel Xeon 32-bit processors with at least 2GB of RAM, running Red Hat Linux.

3 WestGrid orcinus cluster

The WestGrid orcinus cluster is composed of 12 chassis, each containing 16 blades with two compute servers on each blade and each server has two 3.0 GHz Intel Xeon E5450 quad-core processors, with each server sharing 16 GB of RAM, running Red Hat Enterprise Linux Server.

4 Instances

All of our instances are available at:
`http://people.cs.ubc.ca/~davet/papers/sat10-dave-instances.zip`

For our experiments we split the instance sets into halves: a *test* set, and a *training* set. To split the instances we used a stratification strategy to ensure that the test and training set were of approximately equal hardness. As an approximate measure of hardness, we measured the file size of the instance. We first sorted the instances by their file size, and then segmented the instances into pairs of consecutive instances. For each consecutive pair, we randomly placed one instance in the test set, and the other in the training set. Only the instances in the training set were used to determine good parameter settings, and only instances in the test set were used to report experimental results.

5 ParamILS experimental information

All PARAMILS experiments were conducted with the UNIX binary of PARAMILS version 2.3.2. We used the default *FocusedILS* configuration of PARAMILS with settings of (deterministic, overall-obj) = (0, mean10). Because PARAMILS can be very sensitive to the ordering of the instance list, we performed several runs of PARAMILS, each with a randomized instance list, and selected the configuration with the best performance on the training set. To measure this performance we ran each configuration five times on each instance in the training set and measured the median run-length from those five runs, and then measured the mean of those medians. The current PARAMILS software implementation only supports adaptive capping of algorithm runs after a given run-time, not after a given run-length. Since we were interested in optimizing our algorithms in DAVE for run-length performance, but still wanted to take advantage of PARAMILS's excellent adaptive capping feature, we reported the run-length information to PARAMILS as run-time information. Whether or not PARAMILS uses the solution quality to compare unsuccessful runs changed between different PARAMILS versions due to a bug we identified, so to err on the side of caution we included it in our run-time as follows. For a run with a run-length of rl with a solution quality of u unsatisfied clauses, we modified the run-length to be $rl + \frac{u}{10\,000}$. Instead of simply reporting the run-length of DAVE as a run-time, we decided to add an additional transformation to provide PARAMILS with a run-time in the same order of magnitude it typically encounters, to avoid introducing any unintended numerical precision errors. We divided the run-length by 10^6, so that one million search steps in DAVE corresponded to one second in PARAMILS. We had to use a *wrapper script* around DAVE to convert cutoff times received from PARAMILS back to search steps (*i.e.*, multiply by 10^6). For SATENSTEIN-LS and VW2 experiments, the instance cutoff time was 60 seconds, and for DAVE the cutoff was 10 seconds, which is the equivalent of 10^7 search steps. Because PARAMILS measures its total execution time by relying on the reported run-time data, the manner in which we were reporting DAVE performance run-length data to PARAMILS as run-time data was problematic. As a result, we specified a very large amount of cutoff time to PARAMILS and controlled the total amount of CPU time used by PARAMILS through our computation environment.

In our experiments with PARAMILS and VE-SAMPLER, we encountered some difficulties that we believe were caused by the very large parameter space of VE-SAMPLER. Due to Ruby's overhead in some data structures, we observed that the binary Ruby implementation of PARAMILS could consume a large amount of RAM (over 1 GB), and as a consequence would be automatically terminated in our computation environment. In addition, we observed that some of the PARAMILS runs would stagnate, not improving over the initial (default) configuration despite parallel runs achieving great improvement. As a result of these two observations, we used an *iterative* strategy, where instead of executing PARAMILS for some amount of time t, we executed k iterations of PARAMILS, each with time t/k. For each iteration, we would use the best configuration from the previous iteration as the default configuration (see Section 5.4).

5.1 VW2

The possible configurations for each parameter of VW2 were:

```
s {1, 0.33, 0.1, 0.033, 0.01, ... 0.000033, 0.00001, 0}
c {1, 0.33, 0.1, 0.033, 0.01, ... 0.00000033, 0.0000001, 0}
wp {0.05, 0.1, 0.15, 0.2, 0.25, 0.3, 0.35, 0.4, 0.45, 0.5}
```

We ran PARAMILS 10 times on the UBC arrow cluster for 24 hours. The best VW2 configuration found by PARAMILS for CBMC is $(s, c, wp) = (0, 0.01, 0.2)$. The best VW2 configuration found by PARAMILS for SWV is $(s, c, wp) = (0, 0.1, 0.05)$.

5.2 VW2+VE

The possible configurations for each parameter of VW2+VE were:

```
w {0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 20, 25, 30, 35,
   40, 45, 50, 60, 70, 80, 90, 100, 110, 120, 130, 140, 150, 175, 200}
c {0, 0.05, 0.1, 0.15, 0.2, 0.25, 0.3, 0.35, 0.4, 0.45, 0.5, 0.55, 0.6,
   0.65, 0.7, 0.75, 0.8, 0.85, 0.9, 0.95, 1.0, 1.05, 1.1, 1.15, 1.2,
   1.25, 1.3, 1.35, 1.4, 1.45, 1.5, 1.6, 1.7, 1.8, 1.9, 2.0, 2.25, 2.5,
   2.75, 3, 3.5, 4, 4.5, 5, 5.5, 6, 6.5, 7, 8, 9, 10, 15, 20, 25, 30,
   35, 40}
a {0.125, 0.25, 0.5, 1, 2, 4, 8}
```

The value of wp is calculated from w as $\frac{1}{1+w}$. We ran PARAMILS 20 times on the WestGrid glacier cluster for 24 hours. The best VW2+VE configuration found by PARAMILS for CBMC is $(c, a, wp) = (0.95, 8, 0.05)$.

5.3 WALKSAT+VE

For each term in WALKSAT+VE, the possible configurations for the parameters c, a and w (wp) were the same as for VW2+VE. For the normalizations, the possible configurations for make and $\mathsf{relMake}$ were:

$$\|\mathsf{p}\|_{\text{flat}} = \frac{\mathsf{p} - \min(\mathsf{p})}{\max(\mathsf{p}) - \min(\mathsf{p})} \tag{1}$$

$$\|\mathsf{p}\|_{\max} = \frac{\mathsf{p}}{\max(\mathsf{p})} \tag{2}$$

$$\|\mathsf{p}\|_{\text{sum}} = \frac{\mathsf{p}}{\text{sum}(\mathsf{p})} \tag{3}$$

and the possible configurations for break and $\mathsf{relBreak}$ included all of the above normalizations in the form of $(1 - \|\mathsf{p}\|)$, as well as:

$$\|x\|_{\text{-max}} = \frac{\max(x) + \min(x) - x}{\max(x)} \tag{4}$$

We ran PARAMILS 20 times on the WestGrid glacier cluster for 96 hours. The best WALKSAT+VE configuration found by PARAMILS for CBMC is $(wp) = (0.5)$ and the scoring function is:

$$\begin{array}{c} 1.05 \cdot \left(\|\mathsf{make}\|_{\text{flat}}\right)^{8} + 1.35 \cdot \left(\|\mathsf{relMake}\|_{\text{flat}}\right)^{4} \\ +8 \cdot \left(1 - \|\mathsf{break}\|_{\max}\right)^{1/2} + 2.25 \cdot \left(1 - \|\mathsf{relBreak}\|_{\max}\right)^{4} \end{array} . \tag{5}$$

5.4 VE-SAMPLER

For VE-SAMPLER, the possible configurations for the weight of each sub-controller (w), the exponents in the VEs (a) and the co-efficients of the clw function (c) were:

```
w {0, 1, 1.5, 2, 2.5, 3, 4, 5, 7.5, 10, 12.5, 15, 20,
   25, 30, 40, 50, 75, 100}
a {0.0625, 0.125, 0.25, 0.5, 1, 2, 4, 8, 16}
c {0, 0.1, 0.25, 0.5, 0.667, 0.8, 0.9, 0.95, 1, 1.05,
   1.1, 1.25, 1.5, 1.75, 2, 2.5, 3, 3.5, 4, 5, 7.5, 10}
```

The possible configurations for the properties (or ratio of properties) used in the VEs are listed in the paper. For each property, we allowed only two possible normalizations, depending on whether or not the property is a maximal or minimal property (see the paper). The possible normalizations were $\|\mathsf{p}\|_{\text{flat}}$, $\|\mathsf{p}\|_{\max}$, $(1 - \|\mathsf{p}\|_{\text{flat}})$ or $\|\mathsf{p}\|_{\text{-max}}$ (see Section 5.3 above). We ran PARAMILS for 4 iterations, where for each iteration we ran PARAMILS 40 times for 24 hours. For CBMC we used the WestGrid glacier cluster, and for SWV we used the WestGrid orcinus cluster. The best VE-SAMPLER configuration found by PARAMILS on CBMC is:

$$\begin{array}{ll} w_1 = 3 & e_1 = \mathsf{freebie} \\ w_2 = 30 & e_2 = \left(\|\mathsf{break}\|_{\text{-max}}\right)^{1/4} + \text{clw}(0, 0, 2) \cdot \left(1 - \|\mathsf{relBreak}\|_{\text{flat}}\right)^{16} \\ w_3 = 50 & e_3 = \left(\|\mathsf{relMake}\|_{\max}\right) + \text{clw}(3.5, 0.9, 5) \cdot \left(\|\mathsf{age}'\|_{\max}\right)^{8} \\ w_4 = 3 & e_4 = \left(\|\mathsf{make}\|_{\text{flat}}\right) + \text{clw}(0.25, 3.5, 1.05) \cdot \left(\|\mathsf{flips}\|_{\text{-max}}\right)^{1/8} \\ w_5 = 30 & e_5 = \left(\|\mathsf{relMake}\|_{\text{flat}}\right)^{2} + \text{clw}(0.25, 0.25, 3) \cdot \left(\|\mathsf{age}'\|_{\text{flat}}\right) \\ w_6 = 1 & e_6 = \left(\|\mathsf{make}\|_{\max}\right)^{8} + \text{clw}(0.95, 3.5, 0.5) \cdot \left(\|\mathsf{age}\|_{\max}\right) \end{array} \tag{6}$$

The best VE-SAMPLER configuration found by PARAMILS on swv (partial) is:

$$\begin{aligned}
w_1 &= 3 & e_1 &= \textsf{freebie} \\
w_2 &= 15 & e_2 &= (1 - \|\textsf{break}\|_{\text{flat}})^{1/16} + \mathrm{clw}(0, 0.25, 1) \cdot (\|\textsf{make}\|_{\text{max}})^{1/2} \\
w_3 &= 50 & e_3 &= (\|\textsf{break}\|_{\text{-max}})^{16} + \mathrm{clw}(0.1, 5, 0.25) \cdot (\|\textsf{flips}\|_{\text{-max}})^{1/2} \\
w_4 &= 50 & e_4 &= (\|\textsf{break}\|_{\text{-max}})^{1/16} + \mathrm{clw}(0.1, 3.5, 1.75) \cdot (\|\textsf{flips}\|_{\text{-max}})^{1/16} \\
w_5 &= 3 & e_5 &= (\|\textsf{relBreak}\|_{\text{-max}}) + \mathrm{clw}(1, 5, 7.5) \cdot (1 - \|\textsf{flips}\|_{\text{flat}})^{1/16} \\
w_6 &= 5 & e_6 &= (\|\textsf{make}\|_{\text{flat}})^{1/2} + \mathrm{clw}(2.5, 0.1, 1.05) \cdot (\|\textsf{flips}\|_{\text{-max}})
\end{aligned} \tag{7}$$

5.5 SATENSTEIN

For SATENSTEIN-LS, we used two PARAMILS configuration files provided by the SATENSTEIN-LS authors. We ran PARAMILS 40 times (20 times for each configuration file) on the WestGrid orcinus cluster for 96 hours. The configuration of SATENSTEIN-LS found by PARAMILS on swv (partial) is:

```
-adaptive 0 -adaptivenoisescheme 1 -adaptiveprom 0
-adaptpromwalkprob 0 -adaptwalkprob 0 -alpha 1.066 -c 0.00001
-clausepen 1 -decreasingvariable 3 -dp 0.05 -heuristic 2
-maxinc 20 -novnoise 0.5 -performalternatenovelty 1
-performrandomwalk 1 -pflat 0.05 -phi 5 -promdp 0.05
-promisinglist 0 -promnovnoise 0.5 -promphi 5 -promtheta 6
-promwp 0.01 -ps 0 -randomwalk 4 -rdp 0.05 -rfp 0.15 -rho 0.8
-rwp 0.1 -rwpwalk 0.05 -s 0.001 -sapsthresh -0.1
-scoringmeasure 3 -selectclause 1 -singleclause 0
-smoothingscheme 1 -tabu 5 -tabusearch 0 -theta 6
-tiebreaking 2 -updateschemepromlist 3 -varinfalse 1 -wp 0.05
-wpwalk 0.7
```