Detailed SATzilla Results
from the Data Analysis Track of the 2011 SAT Competition

Lin Xu, Frank Hutter, Holger Hoos, and Kevin Leyton-Brown

University of British Columbia,
201-2366 Main Mall, Vancouver, BC, Canada

In previous years, we submitted SATzilla as a solver to the SAT competition, showing that portfolio-based methods can often outperform their constituent solvers. Having established that point, we decided not to compete with solver authors in the SAT competition this year, to avoid discouraging new work on solvers. Instead, we entered SATzilla in the new "analysis track", which allows us to demonstrate SATzilla's performance while keeping the focus where it belongs: on the solvers inside the portfolio. This is important because in some cases, component solvers that make a big difference aren't strong solvers in the "winner take all" sense; analyzing the choices made by a solver like SATzilla can identify such contributions. This document gives a detailed description of that analysis.

1 Experimental setup

Solvers For each category, we constructed portfolios and oracles (SATzilla with a perfect selector) using all non-portfolio, non-parallel solvers from Phase 2 of the competition as component solvers. We compared these to the "virtual best solver" (VBS): an oracle that magically chooses the best solver for each instance and considers all competition entrants (including parallel solvers, solvers we discarded as being unhelpful to SATzilla, etc). The VBS does not represent the current state of the art, since it cannot be run on new instances; nevertheless, it serves as an upper bound on the performance that any portfolio-based selector could achieve. We also compared to the winners of each category (including other portfolio-based solvers). Note that our results cannot be used to determine which portfolio building procedure (e.g., SATzilla, 3S, or ppfolio) is the best, since our SATzilla portfolios had access to data and solvers unavailable to portfolios that competed in the solver track. However, since we combine the most advanced SAT solvers using a tried-and-tested portfolio building procedure, we believe that the resulting portfolios either represent or at least closely approximate the best performance reachable by currently available methods.

Table 1 gives the total number of solvers used for each category of the competition.

Data Set	**Application**	**Crafted**	**Random**
Number of all solvers (used in VBS)	31	25	17
Number of candidates (used in SATzilla and Oracle)	18	15	9

Table 1. *The number of solvers from Phase 2 of the 2011 SAT Competition, and the number of solvers (excluding portfolio-based and multi-threaded procedures) used for building SATzilla 2011 (and the Oracle). See result tables for the solver names.*

Features We used 115 features similar to those used by SATzilla in the 2009 SAT Competition, which fall into the following 9 categories: problem size, variable graph, clause graph, variable-clause graph, balance, proximity to Horn formula, local search probing, clause learning, and survey propagation. Feature computation took on average 31.4 CPU seconds per instance on Random, 51.8 CPU seconds on Crafted, and 158.5 CPU seconds on Application.

Methods We used the new SATzilla procedure (based on cost-sensitive classification models).[1]We used 10-fold cross validation to obtain an unbiased estimate of SATzilla's performance. First, we eliminated all

[1] Please see L. Xu, F. Hutter, H.H. Hoos, and K. Leyton-Brown. Hydra-MIP: Automated algorithm configuration and selection for mixed integer programming. In *RCRA workshop on Experimental Evaluation of Algorithms for Solving Problems with Combinatorial Explosion at IJCAI 2011*, 2011.

instances that could not be solved by any candidate solver (we denote those instances as U). Then, we randomly partitioned the remaining instances (denoted S) into 10 disjoint sets. Treating each of these sets in turn as the test set, we constructed SATzilla using the other 9 sets as training data, and measured SATzilla's runtime on the test set. Finally, we computed SATzilla's average performance across the 10 test sets.

To evaluate how important each solver was for SATzilla, for each category we quantified the percentage of instances solved by each solver during SATzilla's presolving (Pre1 or Pre2), backup, and main stages. Note that our use of cross-validation means that we constructed 10 different SATzilla portfolios using 10 different subsets ("folds") of instances. These 10 portfolios can be qualitatively different (e.g., selecting different presolvers); we report aggregates over the 10 folds. We also quantified the *omission cost* of each candidate solver, defined as the reduction in the percentage of instances SATzilla was able to solve when the given solver was excluded.

Data Runtime data was provided by the organisers of the 2011 SAT competition. All feature computations were performed by Daniel Le Berre on a quad core computer with 4GB of RAM and running Linux, using our code. Four instances (from the Crafted category) out of 1200 had no feature values due to a database problem caused by duplicated file name[2]. We treated these instances as timeouts for SATzilla, thus obtaining a lower bound on SATzilla's performance.

[2] Four pairs of instances in the Crafted category had identical names. Since the database for recording feature values used file name as the primary key, the feature values for four instances could not be written into the database. We sent a request to Daniel Le Berre for re-collecting these feature values, but have not yet received them.

2 Results for Category Application

Solver	Virtual best solver	Oracle	SATzilla 2011 (Application)	Gold medalist: Glucose 2
Average Runtime (CPU seconds)	1104	1138	1685	1856
Solved Percentage	84.7%	84.3%	75.3%	71.7%

Table 2. *Comparison of SATzilla 2011 (Application) with virtual best solver, Oracle (SATzilla with perfect selector), and the gold medalist (Glucose 2) on Application; Timeout runs are counted as the cutoff of 5000 CPU seconds.*

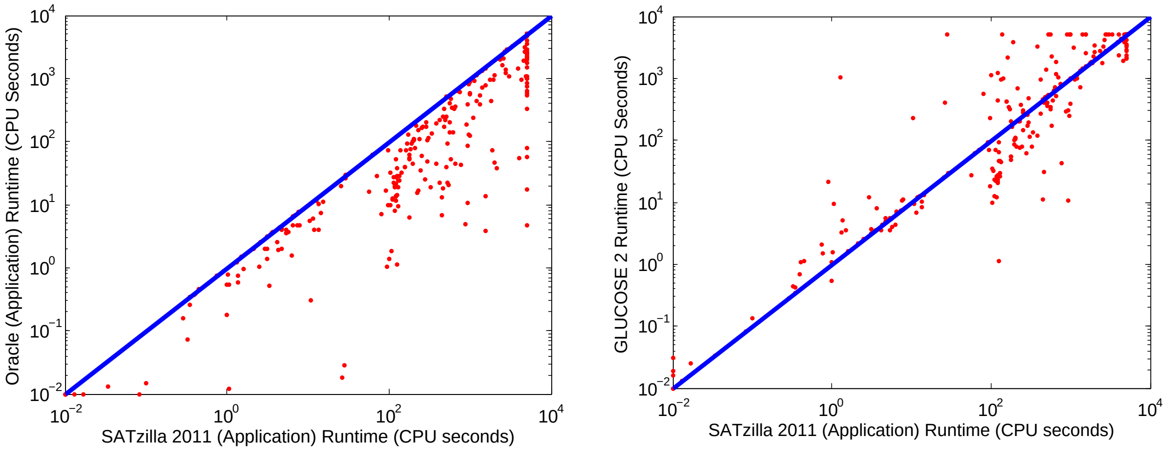

Fig. 1. *Comparison of SATzilla 2011 (Application) with Oracle (SATzilla with perfect selector) [left], and the gold medalist (Glucose 2) [right] on Application; each point corresponds to one SAT instance.*

Solver	Used as Backup	Applied (Solved)	Used as Presolver 1	Solved	Used as Presolver 2	Solved	Picked by Model (Solved)	Cost of Omission
Glucose 2	10/10	10.3% (8.7%)	3/10	6.3%	-	-	9.9% (9.5%)	0.4%
Glueminisat 2.2.5	-	- (-)	5/10	13.4%	-	-	12.7% (9.9%)	0.8%
QuteRSat	-	- (-)	-	-	-	-	12.7% (11.1%)	0.8%
Precosat 236	-	- (-)	-	-	-	-	5.5% (4.7%)	0.4%
EBGlucose 1.0	-	- (-)	1/10	2.8%	-	-	1.9% (1.6%)	0.8%
CryptoMiniSat	-	- (-)	-	-	-	-	3.6% (3.6%)	0.4%
Minisat psm	-	- (-)	1/10	2.8%	-	-	0.4% (0.4%)	1.2%
MPhaseSAT64	-	- (-)	-	-	-	-	3.6% (2.8%)	1.6%
Lingeling 587f	-	- (-)	-	-	-	-	2.4% (2.4%)	0.8%
Contrasat	-	- (-)	-	-	-	-	2.4% (2.0%)	1.2%
Minisat 2.2.0	-	- (-)	-	-	-	-	2.0% (2.0%)	-0.4%
LR GL SHR	-	- (-)	-	-	-	-	2.0% (1.6%)	0.8%
RestartSAT B95	-	- (-)	-	-	-	-	1.9% (1.2%)	0.4%
Rcl	-	- (-)	-	-	-	-	1.2% (1.2%)	0.4%
MiniSAT 2.2.0 agile	-	- (-)	-	-	-	-	1.6% (0.8%)	0.4%
Cir minisat (simp)	-	- (-)	-	-	-	-	0.8% (0.8%)	0.0%

Table 3. *Solvers used in SATzilla 2011 (Application) on Application (after having dropped instances that could not be solved by any candidate). Omission cost for a solver s is negative if SATzilla performed* better *without s. (Usually, SATzilla's solver subset selection phase recognizes that fact and drops s, but this process is not perfect when the training set is too small).*

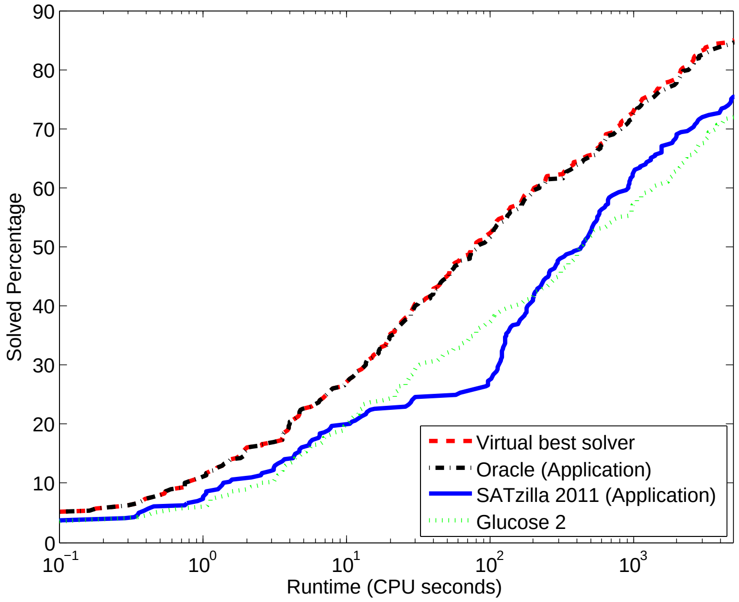

Fig. 2. *Runtime CDF for SATzilla 2011 (Application), virtual best solver, Oracle (SATzilla with perfect selector), and the gold medalist (Glucose 2) on Application.*

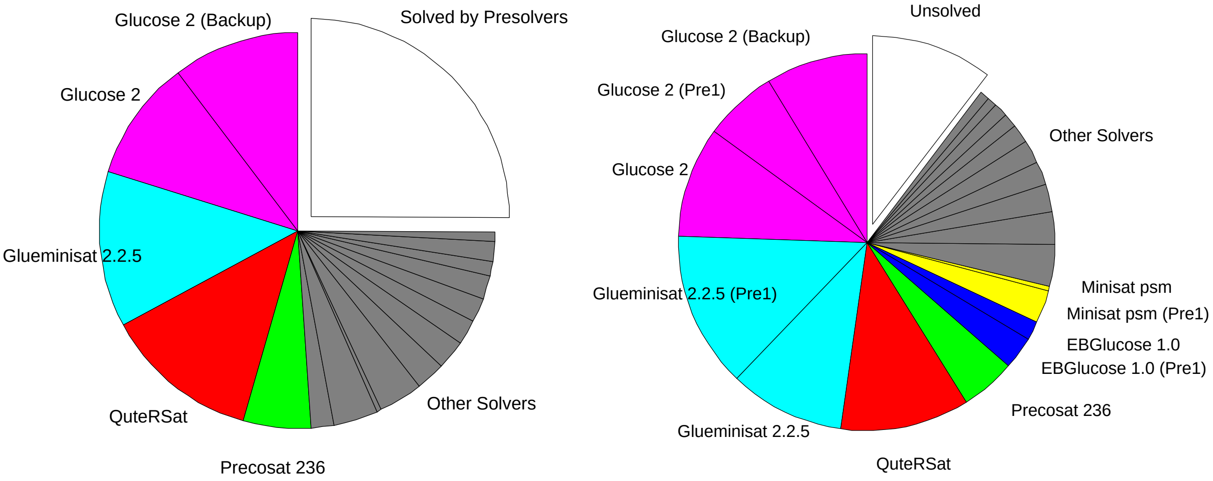

Fig. 3. *Left: solver selection frequencies for SATzilla 2011 (Application); Right: the fraction of instances solved by pre-solvers, the backup solver, and candidate solvers. We only show names for solvers that were picked more than 5% of the time. Data set: Application.*

3 Results for Category Crafted

Solver	Virtual best solver	Oracle	SATzilla 2011 (Crafted)	Gold medalist: 3S
Average Runtime (CPU seconds)	1542	1667	2096	2602
Solved Percentage	76.3%	73.7%	66.0%	54.3%

Table 4. *Comparison of SATzilla 2011 (Crafted) with virtual best solver, Oracle (SATzilla with perfect selector), and the gold medalist (3S) on Crafted; Timeout runs are counted as the cutoff of 5000 CPU seconds.*

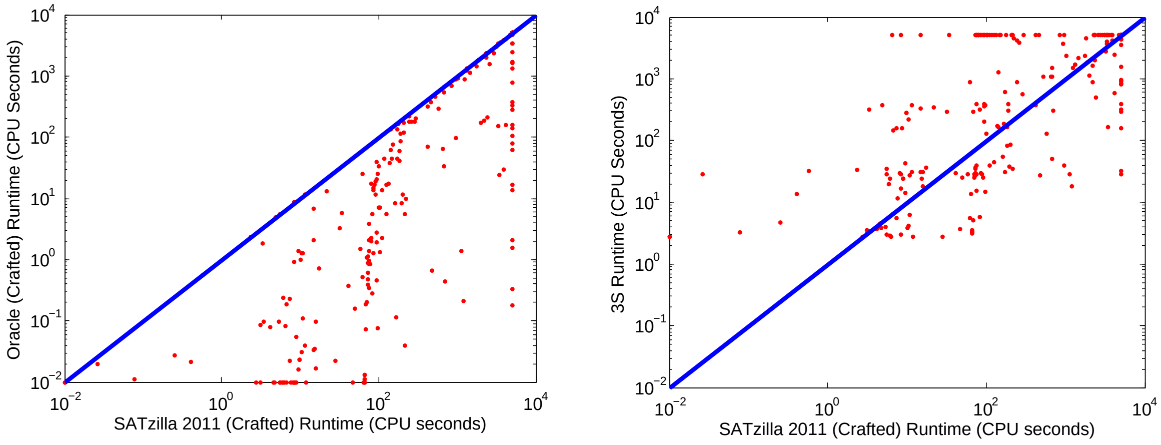

Fig. 4. *Comparison of SATzilla 2011 (Crafted) with Oracle (SATzilla with perfect selector) [left], and the gold medalist (3S) [right] on Crafted; each point corresponds to one SAT instance.*

Solver	Used as Presolver 1	Solved	Used as Presolver 2	Solved	Picked by Model (Solved)	Cost of Omission
Sattime	2/10	6.4%	-	-	15.5% (12.8%)	3.6%
Sol	-	-	1/10	1.4%	13.7% (13.7%)	8.1%
Clasp 2.0 R4092	-	-	-	-	17.4% (15.5%)	2.7%
PicoSAT 941	-	-	-	-	10.1% (10.1%)	0.5%
Clasp 1.2.0 SAT09	-	-	-	-	7.8% (6.9%)	1.4%
QuteRSat	-	-	-	-	6.9% (6.4%)	1.4%
Sattime+	-	-	-	-	6.4% (5.9%)	1.4%
MPhaseSAT	-	-	-	-	5.0% (4.6%)	3.6%
CryptoMiniSat	-	-	-	-	3.2% (2.7%)	0.5%
RestartSAT B95	-	-	-	-	2.7% (1.4%)	1.4%
SApperloT2010	-	-	-	-	1.4% (1.4%)	1.8%
Glucose 2	-	-	-	-	0.5% (0.5%)	3.6%
JMiniSat 2011	-	-	-	-	0.5% (0.5%)	0.9%

Table 5. *Solvers used in SATzilla 2011 (Crafted) on Crafted (after having dropped instances that could not be solved by any candidate). Omission cost for a solver s is negative if SATzilla performed* better *without s. (Usually, SATzilla's solver subset selection phase recognizes that fact and drops s, but this process is not perfect when the training set is too small).*

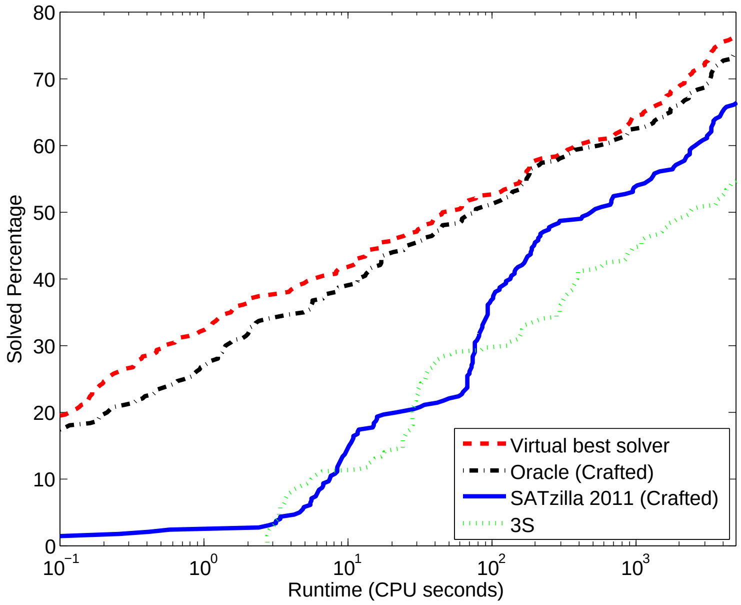

Fig. 5. *Runtime CDF for SATzilla 2011 (Crafted), virtual best solver, Oracle (SATzilla with perfect selector), and the gold medalist (3S) on Crafted.*

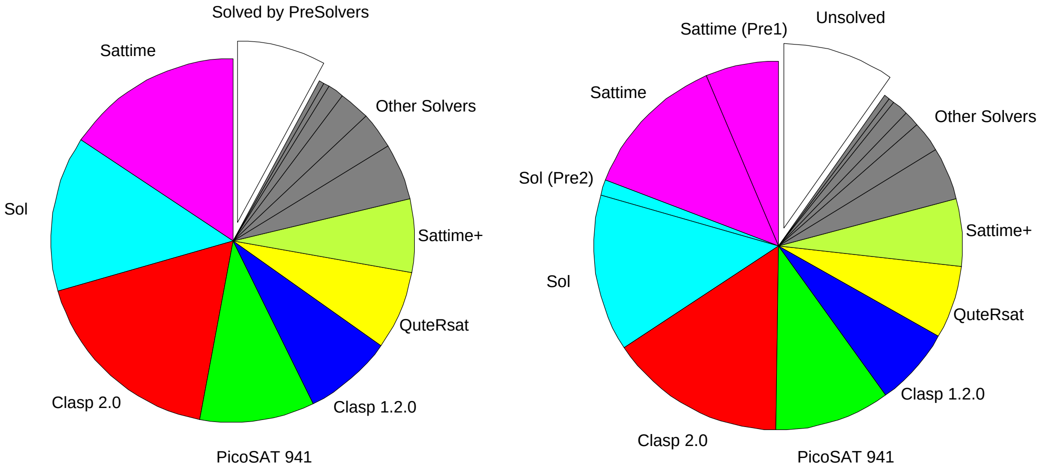

Fig. 6. *Left: solver selection frequencies for SATzilla 2011 (Crafted); Right: the fraction of instances solved by presolvers, the backup solver, and candidate solvers. We only show names for solvers that were picked more than 5% of the time. Data set: Crafted.*

4 Results for Category Random

Solver	Virtual best solver	Oracle	SATzilla 2011 (Random)	Gold medalist: 3S
Average Runtime (CPU seconds)	1074	1087	1172	1836
Solved Percentage	82.2%	82.0%	80.8%	68.0%

Table 6. *Comparison of SATzilla 2011 (Random) with virtual best solver, Oracle (SATzilla with perfect selector), and the gold medalist (3S) on Random; Timeout runs are counted as the cutoff of 5000 CPU seconds.*

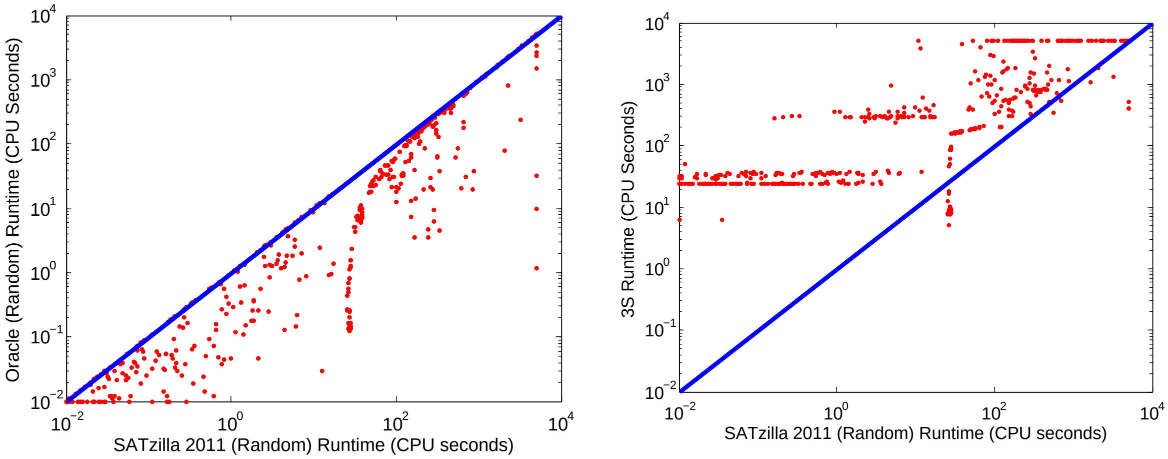

Fig. 7. *Comparison of SATzilla 2011 (Random) with Oracle (SATzilla with perfect selector) [left], and the gold medalist (3S) [right] on Random; each point corresponds to one SAT instance.*

Solver	Used as Presolver 1	Solved	Used as Presolver 2	Solved	Picked by Model (Solved)	Cost of Omission
EagleUP	10/10	47.6%	-	-	4.5% (3.7%)	2.2%
Sparrow2011	-	-	-	-	20.3% (20.3%)	4.9%
March rw	-	-	-	-	20.1% (19.9%)	0.0%
March hi	-	-	-	-	5.3% (5.1%)	0.2%
Gnovelty+2	-	-	-	-	0.8% (0.8%)	0.4%
MPhaseSAT	-	-	-	-	0.4% (0.4%)	0.2%
Sattime	-	-	-	-	0.4% (0.4%)	0.6%
Adaptg2wsat2011	-	-	-	-	0.4% (0.2%)	-0.4%
TNM	-	-	-	-	0.2% (0.2%)	0.4%

Table 7. *Solvers used in SATzilla 2011 (Random) on Random (after having dropped instances that could not be solved by any candidate). Omission cost for a solver s is negative if SATzilla performed* better *without s. (Usually, SATzilla's solver subset selection phase recognizes that fact and drops s, but this process is not perfect when the training set is too small).*

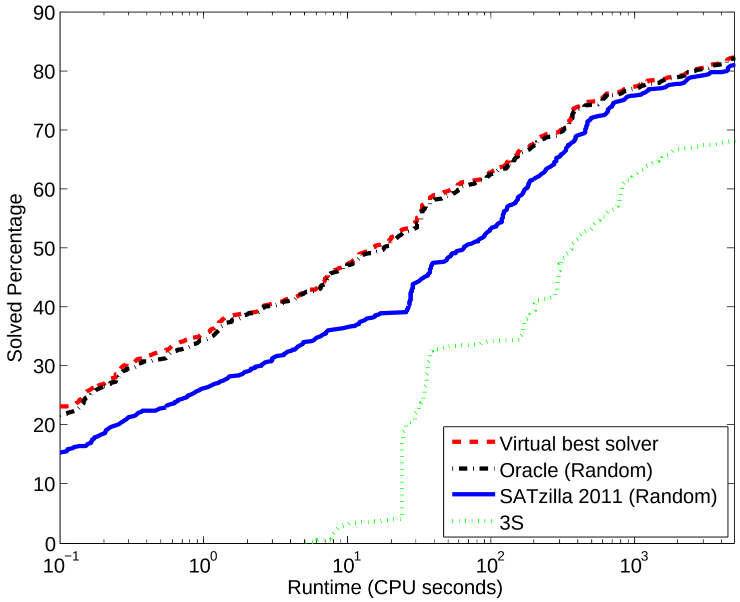

Fig. 8. *Runtime CDF for SATzilla 2011 (Random), virtual best solver, Oracle (SATzilla with perfect selector), and the gold medalist (3S) on Random.*

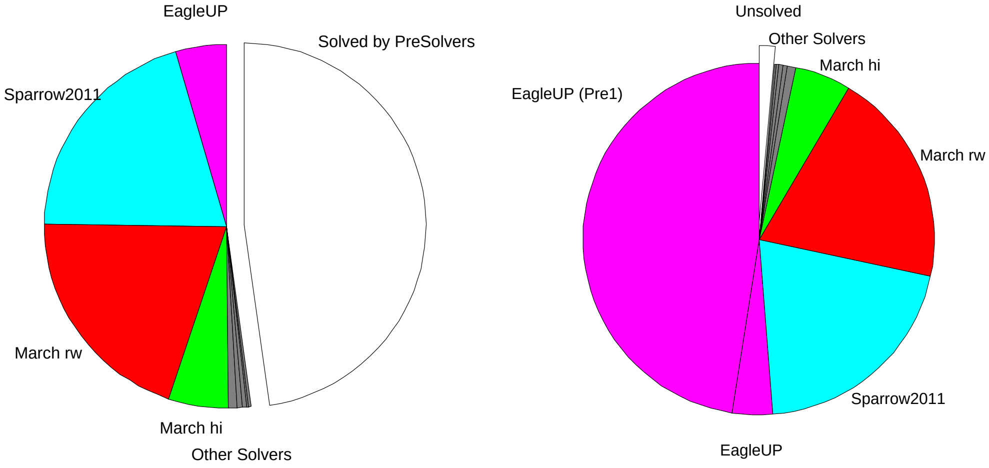

Fig. 9. *Left: solver selection frequencies for SATzilla 2011 (Random); Right: the fraction of instances solved by presolvers, the backup solver, and candidate solvers. We only show names for solvers that were picked more than 5% of the time. Data set: Random.*