Which system should I buy?
A case study about the QBF solvers competition

Cristiano Ghersi and Luca Pulina and Armando Tacchella
DIST - Università di Genova - Viale Causa 15 - 16145 Genova (Italy)
ghersi@star.dist.unige.it - Luca.Pulina@unige.it - Armando.Tacchella@unige.it

Abstract

Systems competitions play a fundamental role in the advancement of the state of the art in several automated reasoning fields. The goal of such events is to answer the question: “Which system should I buy?”. Usually the answer comes as the byproduct of a ranking obtained by considering a pool of problem instances and then aggregating the performances of the systems on each member of the pool. Empirical scoring is the most common ranking method in automated reasoning systems competitions, whereby a tournament-like procedure is used to assign bonuses and penalties to each system according to various performance indicators. Statistical testing is another possible approach, whereby the null hypothesis of equal performances is tested against the alternative hypothesis of significant difference in performances using a precise mathematical formulation. This paper provides a comparison between the two approaches using the 2005 comparative evaluation of solvers for quantified Boolean formulas as a case study.

Introduction

Systems competitions play a fundamental role in the advancement of the state of the art in several automated reasoning fields. A non-exhaustive list of such events includes the CADE ATP System Competition (CASC) (Sutcliffe & Suttner 2007) for theorem provers in first order logic, the SAT Competition (Le Berre & Simon 2007) for propositional satisfiability solvers, the International Planning Competition (see, e.g., (Long & Fox 2003)) for symbolic planners, the CP Competition (see, e.g., (van Dongen 2005)) for constraint programming systems, the Satisfiability Modulo Theories (SMT) Competition (see, e.g., (Barrett, de Moura, & Stump 2005)) for SMT solvers, and the evaluation of quantified Boolean formulas solvers (QBFEVAL, see (Berre, Simon, & Tacchella 2003; Berre *et al.* 2004; Narizzano, Pulina, & Tacchella 2006b) for previous reports). The main purpose of the above events is to designate a winner, i.e., to answer the question: “Which system should I buy?”. Even if such perspective can be limiting, and the results of automated reasoning systems competitions may provide less insight than controlled experiments in the spirit of (Hooker 1996), there is a general agreement that competitions raise interest in the community and they help to set research challenges for developers and assess the current technological frontier for users. The usual way to designate a winner in competitions is to compute a ranking obtained by considering a pool of problem instances and then aggregating the performances of the systems on each member of the pool. While the definition of performances can encompass many aspects of a system, usually it is the capability of giving a sound solution to a high number of problems in a relatively short time that matters most. Therefore, one of the issues that occurred to us as organizers of QBFEVAL, relates to the procedures used to compute the final ranking of the solvers, i.e., we had to answer the question “Which aggregation procedure is best?”. Indeed, even if the final rankings cannot be interpreted as absolute measures of merit, they should at least represent the relative strength of a system with respect to the other competitors based on the difficulty of the problem instances used in the contest.

In this paper we consider two approaches that can be used to summarize the results of a competition: empirical scoring and statistical testing. Empirical scoring is the most common ranking method, whereby a tournament-like procedure is used to assign bonuses and penalties to each system according to various performance indicators. The main advantages of such procedures are their simplicity and their wide applicability, but they offer no direct way of assessing the quality of the rankings provided. Statistical testing, on the other hand, is less commonly used (see, e.g., (Long & Fox 2003)), usually more complicated, and less widely applicable than empirical scoring. However, statistical testing provides direct means of assessing the result quality, since the null hypothesis of equal performances is tested against the alternative hypothesis of significant difference in performances using a precise mathematical formulation that allows an estimate of the results within some stated confidence level. Using the data of the 2005 comparative evaluation of QBF solvers (QBFEVAL’05 (Narizzano, Pulina, & Tacchella 2006b)) as a case study, we summarize the results presented in (Narizzano, Pulina, & Tacchella 2006c) and (Narizzano, Pulina, & Tacchella 2006a). Our goal is to assess whether the ranking obtained with an empirical scoring methods is compatible with the results obtained by statistical testing, and whether statistical testing can help us to improve the scientific significance of competitions.

Preliminaries

The results of QBFEVAL'05 can be listed in a table RUNS comprised of four attributes: SOLVER, INSTANCE, RESULT, and CPUTIME. The attributes SOLVER and INSTANCE report which solver is run on which instance. RESULT is a four-valued attribute: SAT (resp. UNSAT), i.e., the instance was found satisfiable (resp. unsatisfiable) by the solver, TIME, i.e., the solver exceeded the time limit (900 seconds), and FAIL, i.e., the solver aborted for some reason beyond our control. Finally, CPUTIME reports the CPU time spent by the solver on the given instance. In the analysis herewith presented we used a subset of QBFEVAL'05 RUNS table, including only the solvers that admitted to the second stage of the evaluation, namely QUANTOR, QMRES, SEMPROP, YQUAFFLE, SSOLVE, WALKQSAT, OPENQBF and QBF-BDD, and the QBFs coming from classes of instances having fixed structure (see (Narizzano, Pulina, & Tacchella 2006b) for more details).

The analysis herewith presented rests on the assumption that a table identical to RUNS is the only input required by a scoring method. As a consequence, we do not take into account (i) memory consumption, (ii) correctness of the solution, and (iii) "quality" of the solution. The only measures of merit at our disposal are the number of problems solved and the CPU time of the solvers. Notice that the number of problems solved is correct as long as the CPU time measure used to enforce the time limit is so. Therefore, to ensure accuracy of the empirical scoring methods it is very important to tame potential sources of errors in the CPU time measures. Here we observe that statistical testing naturally obviates to this issue, since the samples are already assumed to be noisy observations of the true (unknown) values.

We conclude our preliminaries by briefly describing the empirical scoring methods used in our analysis. For the sake of comparison, Table 1 shows the total scores earned by QBFEVAL'05 participants according to the methods considered.

Borda's method (Saari 2001) Suppose that n solvers (candidates) and m instances (voters) are involved in the contest. Consider the sorted list of solvers obtained for each instance by considering the value of the CPUTIME field in ascending order. Let $p_{s,i}$ be the position of a solver s ($1 \leq s \leq n$) in the list associated with instance i ($1 \leq i \leq m$). According to Borda's method, each voter's ballot consists of a vector of individual scores given to candidates, where the score $S_{s,i}$ of solver s on instance i is simply $S_{s,i} = n - p_{s,i}$. In cases of time limit attainment or failure, we default $S_{s,i}$ to 0. The score of a candidate, given the individual preferences, is just $S_s = \sum_i S_{s,i}$, and the winner is the solver with the highest score.

CASC (Sutcliffe & Suttner 2007) Using CASC methodology, the solvers are ranked according to the number of problems solved, i.e., the number of times RESULT is either SAT or UNSAT. Under this procedure, solver A is better than solver B, if and only if A is able to solve at least one problem more than B within the time limit. In case of a tie, the solver faring the lowest average on CPUTIME fields over the problems solved is the one which ranks first.

QBF evaluation (Narizzano, Pulina, & Tacchella 2006b) QBFEVAL methodology is the same as CASC, except for the tie-breaking rule, which is based on the sum of CPUTIME fields over the problems solved.

Range voting Again, suppose that n solvers and m instance are involved in the contest and $p_{s,i}$ is obtained as described above for Borda's method. We let the score $S_{s,i}$ of solver s on instance i be the quantity $ar^{n-p_{s,i}}$, i.e., we use a positional scoring rule following a geometric progression with a common ratio $r = 2$ and a scale factor $a = 1$. We consider failures and time limit attainments in the same way (we call this the failure-as-time-limit model in (Narizzano, Pulina, & Tacchella 2006a)), and thus we assume that all the voters express an opinion about all the solvers. The overall score of a candidate is again $S_s = \sum_i S_{s,i}$ and the candidate with the highest score wins the election.

SAT competition (Le Berre & Simon 2007) The last SAT competition uses a *purse-based method*, i.e., the measure of effectiveness of a solver on a given instance is obtained by adding up three purses:

- the solution purse, which is divided equally among all solvers that solve the problem;
- the speed purse, which is divided unequally among all the competitors that solve the problem, first by computing the speed factor $F_{s,i}$ of a solver s on a problem instance i:

  $$F_{s,i} = \frac{k}{1 + T_{s,i}} \qquad (1)$$

  where k is an arbitrary scaling factor (we set $k = 10^4$ according to (Gelder *et al.* 2006)), and $T_{s,i}$ is the time spent by s to solve i; then by computing the speed award $A_{s,i}$, i.e., the portion of speed purse awarded to the solver s on the instance i:

  $$A_{s,i} = \frac{P_i \cdot F_{s,i}}{\sum_r F_{r,i}} \qquad (2)$$

  where r ranges over the solvers, and P_i is the total amount of the speed purse for the instance i.
- the series purse, which is divided equally among all solvers that solve at least one problem in a given series (a series is a family of instances that are somehow related, e.g., different QBF encodings for some problem in a given domain).

The overall ranking of the solvers under this method is obtained by considering the sum of the purses obtained on each instance, and the winner of the contest is the solver with the highest sum.

Schulze's method We denote as such an extension of the method described in Appendix 3 of (Schulze 2003). Since Schulze's method is meant to compute a single overall winner, we extended the method according to its author's suggestions in order to make it capable of generating an overall ranking.

YASMv2 While the aggregation procedures used in CASC and QBF evaluations are straightforward, they do not take into account some aspects that are indeed considered by the purse-based method used in the last SAT competition. On the other hand, the purse-based method used in SAT requires some oracle to assign purses to the problem instances, so the results can be influenced heavily by the oracle. In (Pulina 2006) a first version of YASM was introduced as an attempt to combine the two approaches: a rich method like the purse-based one, but using the data obtained from the runs only. As reported in (Pulina 2006), YASM featured a somewhat complex calculation, yielding unsatisfactory results, particularly in the comparison with the final ranking produced by voting systems. In (Narizzano, Pulina, & Tacchella 2006c) we revised the original version of YASM to make its computation simpler, and to improve its performance using ideas borrowed from voting systems. From here on, we call YASMv2 the revised version, and YASM the original one presented in (Pulina 2006). YASMv2 requires a preliminary classification whereby a hardness degree H_i is assigned to each problem instance i using the same equation as in CASC (Sutcliffe & Suttner 2007) (and YASM):

$$H_i = 1 - \frac{S_i}{S_t} \tag{3}$$

where S_i is the number of solvers that solved i, and S_t is the total number of participants to the contest. Considering equation (3), we notice that $0 \leq H_i \leq 1$, where $H_i = 0$ means that i is relatively easy, while $H_i = 1$ means that i is relatively hard. We can then compute the measure of effectiveness $S_{s,i}$ of a solver s on a given instance i (this definition changes with respect to YASM):

$$S_{s,i} = k_{s,i} \cdot (1 + H_i) \cdot \frac{L - T_{s,i}}{L - M_i} \tag{4}$$

where L is the time limit, $T_{s,i}$ is the CPU time used up by s to solve i ($T_{s,i} \leq L$), and $M_i = min_s\{T_{s,i}\}$, i.e., M_i is the time spent on the instance i by the *SOTA solver* defined in (Narizzano, Pulina, & Tacchella 2006b) to be the ideal solver that always fares the best time among all the participants. The hybridization with voting systems comes into play with the coefficient $k_{s,i}$ which is computed as follows. Suppose that n solvers are participating to the contest. Each instance ranks the solvers in ascending order considering the value of the CPUTIME field. Let $p_{s,i}$ be the position of a solver s in the ranking associated with instance i ($1 \leq p_{s,i} \leq n$), then $k_{s,i} = n - p_{s,i}$. In case of time limit attainment and failure, we default $k_{s,i}$ to 0, and thus also $S_{s,i}$ is 0. The overall ranking of the solvers is computed by considering the values $S_s = \sum_i S_{s,i}$ for all $1 \leq s \leq n$, and the solver with the highest sum wins. We can see from equation (4) that in YASMv2 the effectiveness of a solver on a given instance is influenced by three factors, namely (i) a Borda-like positional weight ($k_{s,i}$), (ii) the relative hardness of the instance ($1 + H_i$), and (iii) the relative speed of the solver with respect to the fastest solver on the instance ($\frac{L-T_{s,i}}{L-M_i}$). Intuitively, coefficient (ii) rewards the solvers that are able to solve hard instances, while (iii) rewards the solvers that are faster than other competitors. The coefficient $k_{s,i}$ has been added to stabilize the final ranking and make it less sensitive to an initial bias in the test set. As we show in the next Section, this combination allows YASMv2 to reach the best compromise among different effectiveness measures.

	CASC/QBF	**SAT**	**YASMv2**	**Borda**	**range voting**	**Schulze**
openQBF	201	62621.96	482.92	436	1682	421
qbfbdd	106	39250.40	363.74	338	3236	273
QMRes	227	173068.18	1505.03	1085	12050	1007
quantor	318	228854.86	2701.35	2019	32393	1824
semprop	289	148690.91	1787.92	1569	18317	1372
ssolve	243	110121.36	1415.36	1286	16038	1135
WalkQSAT	189	87535.25	1090.98	962	11010	791
yQuaffle	250	110257.07	1351.92	1200	11864	1023

Table 1: Scores of QBFEVAL'05 solvers according to the methods considered.

	CASC	**QBF**	**SAT**	**YASM**	**YASMv2**	**Borda**	**r.v.**	**Schulze**
CASC	–	1	0.71	0.86	0.79	0.86	0.71	0.86
QBF		–	0.71	0.86	0.79	0.86	0.71	0.86
SAT			–	0.86	0.86	0.71	0.71	0.71
YASM				–	0.86	0.71	0.71	0.71
YASMv2					–	0.86	0.86	0.86
Borda						–	0.86	1
r. v.							–	0.86
Schulze								–

Table 2: Homogeneity of aggregation procedures.

Empirical Scoring

In this section we summarize the results obtained considering the above mentioned scoring methods and some effectiveness measures introduced in (Pulina 2006) and (Narizzano, Pulina, & Tacchella 2006c) that are meant to show whether the aggregation procedures have some desirable properties, including fidelity and stability with respect to various perturbations that may occur in the test set used for the competition.

Homogeneity

The rationale behind this measure (introduced in (Pulina 2006)) is to verify that, on a given test set, the aggregation procedures considered (i) do not produce exactly the same solver rankings, but, at the same time, (ii) do not yield antithetic solver rankings. Thus, homogeneity is not an effectiveness measure per se, but it is a preliminary assessment that we are performing an apple-to-apple comparison and that the apples are not exactly the same.

Homogeneity is computed as in (Pulina 2006) considering the Kendall rank correlation coefficient τ which is a non-parametric coefficient best suited to compare rankings. τ is computed between any two rankings and it is such that $-1 \leq \tau \leq 1$, where $\tau = -1$ means perfect disagreement, $\tau = 0$ means independence, and $\tau = 1$ means perfect agreement. Table 2 shows the values of τ computed for the aggregation procedures considered, arranged in a symmetric matrix where we omit the elements below the diagonal (r.v. is a shorthand for range voting). Values of τ close to, but not exactly equal to 1 are desirable. Table 2 shows that this is indeed the case for the aggregation procedures considered using QBFEVAL'05 data. Only two couples of methods (QBF-CASC and Schulze-Borda) show perfect agreement, while all the other couples agree to some extent, but still produce different rankings.

Method	Mean	Std	Median	Min	Max	IQ Range	F
QBF	182	7	183	170	192	13	88.54
CASC	182	7	183	170	192	13	88.54
SAT	87250	12520	83262	78532	119780	4263	65.56
YASM	46	2	46	43	51	2	85.38
YASMv2	1257.29	45.39	1268.73	1198.43	1312.72	95.11	91.29
Borda	984	127	982	752	1176	194	63.95
r. v.	12010	5183	12104	5186	21504	8096	24.12

Table 3: Fidelity of aggregation procedures. As far as **SAT** is concerned, the series purse is not assigned.

Fidelity

We introduced this measure in (Narizzano, Pulina, & Tacchella 2006c) to check whether the aggregation procedures under test introduce any distortion with respect to the true merits of the solvers. Our motivation is that we would like to extract some scientific insight from the final ranking of QBFEVAL'06 and not just winners and losers. Of course, we have no way to know the true merits of the QBF solvers: this would be like knowing the true statistic of some population. Therefore, we measure fidelity by feeding each aggregation procedure with "white noise", i.e., several samples of table RUNS filled with random results. In particular, we assign to RESULT one of SAT/UNSAT, TIME and FAIL values with equal probability, and a value of CPUTIME chosen uniformly at random in the interval [0;1]. Given this artificial setting, we know in advance that the true merit of the competitors is approximately the same. A high-fidelity aggregation procedure is thus one that computes approximately the same scores for each solver, and thus produces a final ranking where scores have a small variance-to-mean ratio.

The results of the fidelity test are presented in Table 3 where each line contains the statistics of a aggregation procedure. The columns show, from left to right, the mean, the standard deviation, the median, the minimum, the maximum and the interquartile range of the scores produced by each aggregation procedure when fed by white noise. The last column is our fidelity coefficient F, i.e., the percent ratio between the lowest score (solver ranked last) and the highest one (solver ranked first): the higher the value of F, the more the fidelity of the aggregation procedure. As we can see from Table 3, the fidelity of YASMv2 is better than that of all the other methods under test, including QBF and CASC which are second best, and have higher fidelity than YASM. Notice that range voting, and to a lesser extent also SAT and Borda's methods, introduce a substantial distortion. In the case of range voting, this can be explained by the exponential spread that separates the scores, and thus amplifies even small differences. Measuring fidelity does not make sense in the case of Schulze's method. Indeed, given the characteristics of the "white noise" data set, Schulze's method yields a tie among all the solvers. Thus, checking for fidelity would essentially mean checking the tie-breaking heuristic, and not the main method.

RDT-stability and DTL-stability

Stability on a randomized decreasing test set (RDT-stability), and stability on a decreasing time limit (DTL-stability) have been introduced in (Pulina 2006) to measure how much an aggregation procedure is sensitive to perturbations that diminish the size of the original test set, and how much an aggregation procedure is sensitive to perturbations that diminish the maximum amount of CPU time granted to the solvers, respectively. The conclusion reached in (Narizzano, Pulina, & Tacchella 2006c) are:

- All the aggregation procedures considered are substantially RDT-stable, i.e., a random sample of 151 instances is sufficient for all the procedures to reach the same conclusions that each one reaches on the heftier set of 551 instances used in QBFEVAL'05.
- Decreasing the time limit substantially, even up to one order of magnitude, is not influencing the stability of the aggregation procedures considered, except for some minor perturbations for QBF/CASC, SAT and Schulze's methods. Moreover, independently from the procedure used and the amount of CPU time granted, the best solver is always the same.

Indeed, while the above measures can help us extract general guidelines about running a competition, in our setting they do not provide useful insights to discriminate the relative merits of the procedures.

SBT-stability

Stability on a solver biased test set (SBT-stability) is introduced in (Pulina 2006) to measure how much an aggregation procedure is sensitive to a test set that is biased in favor of a given solver. Let Γ be the original test set, and Γ_s be the subset of Γ such that the solver s is able to solve exactly the instances in Γ_s. Let $R_{q,s}$ be the ranking obtained by applying the aggregation procedure q on Γ_s. If $R_{q,s}$ is the same as the original ranking R_q, then the aggregation procedure q is SBT-stable with respect to the solver s. Notice that, contrarily to what stated in (Pulina 2006), SBT-stability alone is not a sufficient indicator of the capacity of an aggregation procedure to detect the absolute merit of the participants. Indeed, it turns out that a very low-fidelity method such as range voting is remarkably SBT-stable. This because we can raise the SBT-stability of a ranking by decreasing its fidelity: in the limit, an aggregation procedure that assigns fixed scores to each solver, has the best SBT-stability and the worst fidelity. Therefore, an aggregation procedure showing a high SBT-stability is relatively immune to bias in the test set, but it must also feature a high fidelity if we are to conclude that the method provides a good hint at detecting the absolute merit of the solvers.

Figure 1 shows the plots with the results of the SBT-stability measure for each aggregation procedure. The x-axis reports the name of the solver s used to compute the solver-biased test set Γ_s and the y-axis reports the score value. For each of the Γ_s's, we report eight bars showing the scores obtained by the solvers using only the instances in Γ_s. The order of the bars (and of the legend) corresponds to the ranking obtained with the given aggregation procedure on the original test set Γ. As we can see from Figure 1 (top-left), CASC/QBF aggregation procedures are not SBT-stable: for each of the Γ_s, the original ranking is perturbed and the winner becomes s. Notice that on Γ_{QUANTOR}, CASC/QBF yield the same ranking that they output on the complete test set Γ. The SAT competition procedure (Figure 1, top-center)

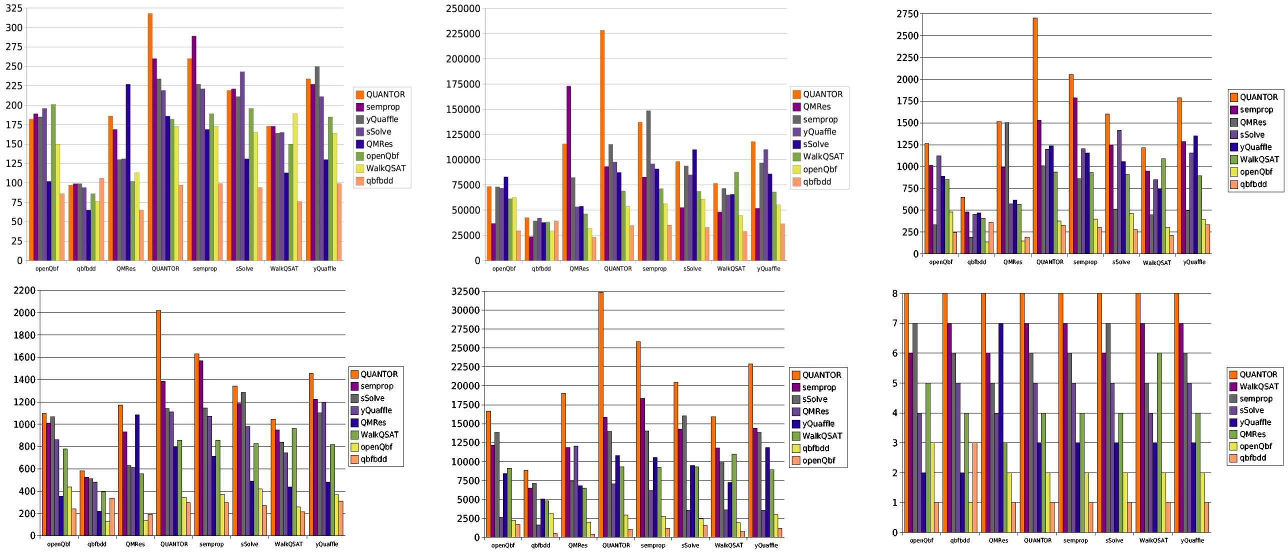

Figure 1: SBT-stability plots; top-row, from left to right, CASC/QBF, SAT, YASMv2; bottom-row, from left to right, Borda, r.v. and Schulze.

is not SBT-stable, not even on the test set biased on its alleged winner QUANTOR. YASMv2 is better than both CASC/QBF and SAT, since its alleged winner QUANTOR is the winner on biased test sets as well. Borda's method (Figure 1, bottom-left) is not SBT-stable with respect to any solver, but the alleged winner (QUANTOR) is always the winner on the biased test sets. Moreover, the rankings obtained on the test sets biased on QUANTOR and SEMPROP are not far from the ranking obtained on the original test set. Also range voting (Figure 1, bottom-center), is not SBT-stable with respect to any solver, but the solvers ranking first and last do not change over the biased test sets and it is true for the Schulze's method (Figure 1, bottom-right) too.

Looking at the results presented above, we can see that YASMv2 performance in terms of SBT stability lies in between classical automated reasoning contests methods and methods based on voting systems. This fact is highlighted in Table 4, where for each procedure we compute the Kendall coefficient between the ranking obtained on the original test set Γ and each of the rankings obtained on the Γ_s test sets, including the mean coefficient observed. Overall, YASMv2 turns out to be, on average, better than CASC/QBF, SAT, and YASM, while it is worse, on average, than the methods based on voting systems. However, if we consider also the results of Table 3 about fidelity, we can see that YASMv2 offers the best compromise between SBT-stability and fidelity. Indeed, while CASC/QBF methods have a relatively high fidelity, they perform poorly in terms of SBT-stability, and SAT method is worse than YASMv2 both in terms of fidelity and in terms of SBT-stability. Methods based on voting systems are all more SBT-stable that YASMv2, but they have poor fidelity coefficients. We consider this good performance of YASMv2 a result of our choice to hybridize classical methods used in automated reasoning contests and methods based on voting systems. This helped us to obtain an aggregation procedure which is less sensitive to bias, and, at the same time, a good indicator of the absolute merit of the competitors.

Statistical Testing

In spite of the results presented insofar, we have no direct means of assessing the significance of the results obtained by an empirical scoring method like YASMv2. Indeed, even if we can do this indirectly using the measures presented in the previous Section, there is no guarantee that the results obtained will apply to a different set of solvers and/or problem instances. On the other hand, if we rephrase the problem in terms of statistical hypothesis testing, then we can check for statistically significant differences in the performances of the solvers and validate our conclusions within some stated confidence level. Let us start by introducing a null hypothesis and an alternative hypotheses that are appropriate in our context. Given any two solvers A and B we can state the:

null hypothesis (H_0), i.e., there are no significant differences in the performances of A with respect to the performances of B; and the

alternative hypothesis (H_1), i.e., there are significant differences in the performances of A with respect to the performance of B.

In the following, let X_A and X_B be the vectors of run-time values associated to solver A and solver B, respectively. Before applying statistical methods to QBFEVAL'05 data, we must decide (i) how to consider missing values, i.e., TIMEOUT and FAIL values, and (ii) which assumptions, if any, can be made about the run-time distributions. The above issues have an impact over the specific method that we can apply to test H_0, because some methods cannot deal with missing values in X_A and X_B seamlessly, and most methods require binding assumptions about the underlying distribution of X_A and X_B.

	CASC/QBF	**SAT**	**YASM**	**YASMv2**	**Borda**	**r. v.**	**Schulze**
OPENQBF	0.43	0.57	0.36	0.64	0.79	0.79	0.79
QBFBDD	0.43	0.43	0.36	0.64	0.79	0.86	0.79
QMRES	0.64	0.86	0.76	0.79	0.71	0.86	0.79
QUANTOR	1	0.86	0.86	0.86	0.93	0.86	0.93
SEMPROP	0.93	0.71	0.71	0.79	0.93	0.86	0.93
SSOLVE	0.71	0.57	0.57	0.79	0.86	0.79	0.86
WALKQSAT	0.57	0.57	0.43	0.71	0.64	0.79	0.79
YQUAFFLE	0.71	0.64	0.57	0.71	0.86	0.86	0.93
Mean	0.68	0.65	0.58	0.74	0.81	0.83	0.85

Table 4: Kendall coefficient between the ranking obtained on the original test set and each of the rankings obtained on the solver-biased test sets.

Considering QBFEVAL'05 data, after removing the instances where all the solvers either fail or reach the time limit, there are two possible models to deal with the remaining missing values: failure-as-time-limit (FAT) model, and time-limit-as-failure (TAF) model. In the FAT model, each time that a solver fails or exceeds the time limit, we default its run time to the time limit. This model (used, e.g., in (Hooker & Vinay 1995)) consistently overestimates the performances of the solvers, but allows the paired comparison of the values in X_A and in X_B. In the TAF model, each time a solver fails or exceeds the time limit, we simply disregard the data point. In this way overestimation does not occur, but since the vectors X_A and X_B may not be equal in length, the paired comparison of run-times is not generally possible.

Given FAT and TAF data models, we may ask whether an underlying normal distribution of run-times can be assumed. If so, well-known classical techniques like t-tests or Analysis of Variance (ANOVA) (Kanji 1999) could be used to test for H_0. Thus, for each solver A, we check X_A under FAT and TAF models using the Shapiro-Wilk (Kanji 1999) test of the null hypothesis that the X_A's are obtained from a normally distributed population. All such tests yield p-values in the order of 10^{-27} for the FAT model, and of 10^{-24} for the TAF model, indicating that it is highly unlikely that the run-time distribution of some solver is anywhere close to normal. Because of this, we must resort to non-parametric tests. Inspired by (Long & Fox 2003), we consider the Wilcoxon signed rank (WSR) test, a non-parametric alternative to the classical paired t-test, whereby the null hypothesis states that X_A and X_B do not differ in a significant way (see, e.g., Ch. 12a of (Lowry 2006)). The WSR test is applicable as long as:

1. the paired values of X_A and X_B are randomly and independently drawn;
2. the dependent variable (i.e., the run-time) is intrinsically continuous; and
3. it makes sense to compare the values in X_A and X_B.

Both FAT and TAF models fulfill the above conditions, but considering the mechanics of the WSR test, we see that it requires the vector $X_A - X_B$ to be computed. Therefore, it can be applied only in the context of the FAT model. In order to cope with the TAF model, we consider the Wilcoxon rank sum test, also known as Mann-Whitney test (see, e.g., Ch. 12a of (Lowry 2006)). Such test, that we call hereafter Mann-Whitney-Wilcoxon (MWW) test, is a non-parametric test of difference between X_A and X_B, and it can accommodate for unequal lengths of X_A and X_B. In particular, MWW is applicable as long as conditions 2 and 3 above hold, and it requires that the values of X_A and X_B are independently drawn: since in our case the data are (positively) dependent, MWW gives approximate, although conservative, solutions. In the following, all the data and the results extrapolated from the WSR and MWW tests are implicitly referred to the FAT and TAF models, respectively.

The results of the pairwise WSR and MWW tests on QBFEVAL'05 data are shown in Table 5. In the Table, for each pair of solvers A (row) and B (column) we report the p-value adjusted for multiple comparisons of the pairwise WSR tests and pairwise MWW tests. Before analyzing the results of Table 5 it is worth mentioning that adjustment of the p-values is necessary, because performing multiple comparisons raises the risk of obtaining positive results just by the effect of chance (a fact that follows from Bonferroni inequalities). We applied Holm's adjustment method that should be more effective than the well known Bonferroni's method. Both methods give strong control on the family wise error rate, i.e., the probability that the number of overall false rejections (false positives) is greater or equal to one. In (Long & Fox 2003) the authors use a different correction, i.e., they compute the overall α_0 using $\alpha_0 = 1 - (1-\alpha)^{(1/n)}$, where α is the confidence level of each test and n is the number of tests performed. However, the latter kind of correction is less generally valid than Bonferroni's (and thus also Holm's) method. With this proviso, since we wish to extrapolate a partial order of the solvers in terms of their relative speed performances at a 99% confidence level, we reject H_0 only when the p-value shown in Table 5 is less than 0.01. By looking at Table 5 we can see that the WSR test finds most pairwise comparisons significant at an *overall* 99% confidence level. On the other hand, the MWW test yields more conservative results, i.e., 9 comparisons that are significant for the WSR test fail to be so for the MWW test, and only 1 comparison that is not significant for the WSR test is indeed significant for the MWW test. In Table 5, we highlight in boldface the cases in which the two tests are found disagreeing about the significance of the difference in terms of performances. Notice that WSR and MWW tests are more sensitive to a consistent, albeit small, difference in performances rather than occasional, albeit large, differences.

In Figure 2, we represent two partial orders of the solvers in terms of their performances derived from QBFEVAL'05 data and the results of Table 5. The partial order on the left of Figure 2 is based on the results of the WSR test, while the one on the right is based on the results of the MWW test. Both partial orders are obtained by drawing an edge from

	OPENQBF		QBFBDD		QMRES		QUANTOR		SEMPROP		SSOLVE		WALKQSAT	
	WSR	MWW	WSR	MWW	WSR	MWW	WSR	MWW	WSR	MWW	WSR	MWW	WSR	MWW
QBFBDD	<0.001	<0.001	–	–	–	–	–	–	–	–	–	–	–	–
QMRES	0.003	<0.001	<**0.001**	**1.000**	–	–	–	–	–	–	–	–	–	–
QUANTOR	<0.001	<0.001	<0.001	<0.001	<0.001	<0.001	–	–	–	–	–	–	–	–
SEMPROP	<0.001	<0.001	<**0.001**	**0.982**	<0.001	0.020	<0.001	0.003	–	–	–	–	–	–
SSOLVE	<0.001	<0.001	<**0.001**	**0.982**	0.059	0.024	<0.001	0.002	0.031	1.000	–	–	–	–
WALKQSAT	<0.001	<0.001	<**0.001**	**0.161**	**0.521**	<**0.001**	<**0.001**	**0.566**	<**0.001**	**0.982**	<0.001	0.982	–	–
YQUAFFLE	<0.001	<0.001	<**0.001**	**1.000**	0.018	0.057	<0.001	<0.001	<**0.001**	**1.000**	0.521	1.000	<**0.001**	**0.646**

Table 5: p-values of Wilcoxon signed rank (WSR) and Mann-Whitney-Wilcoxon (MWW) tests.

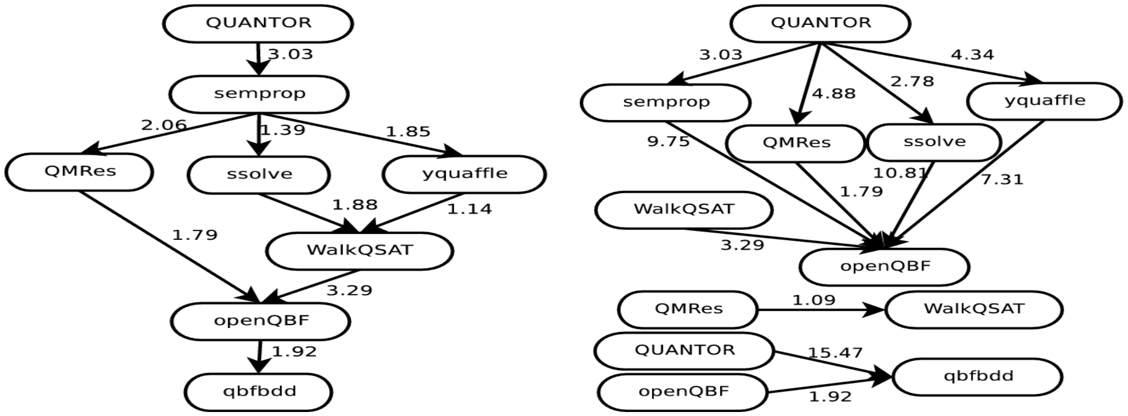

Figure 2: Partial order of the solvers in terms of their relative speed performances extrapolated from the results of Wilcoxon signed rank (left) and Mann-Whiteny-Wilcoxon (right) tests.

A to B whenever there is a significant difference in performances between A and B, while the direction of the edge is obtained considering the pairs (x_A, x_B) such that $x_A \in X_A$ and $x_B \in X_B$ and computing the ratio $R(A, B)$ between (i) the number of pairs such that $x_A < x_B$ and (ii) the number of pairs such that $x_B > x_A$ (ties are thus excluded): if $R(A, B) > 1$, then A is faster than B. In Figure 2 we label each directed edge from A to B with the value of $R(A, B)$ and we omit the links that can be extrapolated by transitive closure. The calculation of R is inspired by the WSR test mechanics and, as such, it is meant to reflect consistent differences in performances rather than occasional large gaps. Looking at Figure 2 we can see that the partial order induced by the results of the WSR test is compatible with the one induced by the results of the MWW test, although the latter is less constrained than the former.

We can now compare the snapshots of QBFEVAL'05 data offered by YASMv2 and the other empirical scoring methods (Table 1) with the ones offered by statistical testing (Figure 2). The first observation is that all the rankings produced by the scoring methods are compatible with the partial orders of Figure 2 with the only exception of SAT, which ranks QMRES above SEMPROP, while there is a consistent and significant difference in performances detected by the WSR test that, together with the value of $R(\text{SEMPROP}, \text{QMRES})$, prompts us otherwise. However, notice that according to the MWW test, such difference is not significant at the 99% confidence level. The second observation is that, looking at Figure 2 we can see that QMRES, SSOLVE and YQUAFFLE are found essentially "incomparable" by the WSR test, and, indeed, the ranking of such solvers is essentially the only part where the scoring methods differ. In particular, if we consider the relative performance index offered by $R(\text{SEMPROP}, B)$, where B is one of QMRES, SSOLVEand YQUAFFLE, we can see that only Borda count and Schulze's method rank the three solvers according to the reverse order of the corresponding edge labels. On the other hand, according to the MWW test (Figure 2, right), also SEMPROP is incomparable to the above three solvers, but SEMPROP always ranks second best according to all the scoring methods (with the above mentioned exception of SAT). Finally, let us consider the total orders extracted by the partial ones of Figure 2 by proceeding top-down and breaking the ties considering the edge labels in reverse order. If we compare the rankings thereby obtained with those resulting from Table 1 using the Kendall coefficient, then we can observe the following. Comparing WSR- and MWW-based rankings yields $\tau = 0.93$, an almost perfect agreement tainted only by the different classification of SEMPROP and SSOLVE. Considering the empirical scoring methods, it turns out that WSR-based ranking yields $\tau = 1$ in the case of Borda count and Schulze's method, and $\tau \neq 1$ in all the other cases, with YASMv2 being the closest (together with range voting) at $\tau = 0.86$. In the case of MWW-based ranking, $\tau \neq 1$ for all the empirical scoring methods considered: YASMv2, with $\tau = 0.79$ is closer than both SAT ($\tau = 0.64$) and CASC/QBF ($\tau = 0.76$), but range voting ($\tau = 0.79$), Schulze's method and Borda count (both at $\tau = 0.93$) are even closer.

Discussion, conclusions and future work

In this paper we have considered the problem of obtaining a fair ranking of the competitors in an automated reasoning contest. We noticed that the problem is ill-posed, since we are comparing systems (not algorithms), and we often have a limited control over the kind of problems that we may use in a competition. Finally, since most automated reasoning problems are *at-least* NP-hard (if at all decidable), it is quite difficult – if at all possible – to describe the distribution of the problem instances using some well-behaved analytical function like, e.g., Gaussian, uniform, or Poisson distributions. This means that, even under the unlikely hypotheses that competitors could submit algorithms instead of systems, and that a high degree of control could be reached on the experimental setup, we would probably be unable to frame a competition using classical statistical methods. On the other hand, the tournament-like aggregation procedures that are normally used in automated reasoning contests, are usually adopted without even asking the question of whether the results they provide are fair, relevant and adequate.

We considered two alternatives to the classic tournament-like procedures. The first is to use some aggregation procedure borrowed from the vast (see (Arrow, Sen, & Suzumura 2002)) literature on social choice, i.e., use some kind of voting system. Voting systems do not require specific assumptions about voters (systems) and solvers (candidates). On the contrary, they are specifically designed to withstand attempts to break them, by voters trying to circumvent the rules. At the same time, a solid mathematical theory has been developed to study the problems of fairness and adequacy of such systems. The second is to use non-parametric, i.e., distribution-free, statistical testing. Non-parametric testing is more complex and, in general, less robust than classical statistical testing, but it can provide with useful results even when the data are scarce or when assumptions about the distributions underlying data cannot be made.

Our empirical results show that a tournament like aggregation procedure like YASM (Pulina 2006) can be improved by borrowing ideas from voting systems. The improved YASMv2 has interesting stability and fidelity properties, and it is also able to "rival" with statistical testing, insofar it is able to detect differences in the performances of the solvers that revealed to be almost always statistically significant under stringent conditions. Indeed, we decided to use YASMv2 (Narizzano, Pulina, & Tacchella 2006c) for the first two competitive evaluations of QBF solvers, QBFEVAL'06 and QBFEVAL'07 (Giunchiglia, Narizzano, & Tacchella 2001). The validity of YASMv2 as an aggregation procedure, however, rests on few assumptions that may not be adequate in other competitions, e.g., we do not take into account the quality of the solution, which is indeed quite important in a scheduling competition.

Summing up, our results clearly indicate that asking the question "Which system should I buy?" has no trivial answer, but that both voting systems and non-parametric statistics can be used to improve the quality of the final answer. One possible development along this line, which we are currently studying, is to leverage statistical testing by scoring several bootstrap replicas of the original test set, i.e., test sets obtained by sampling the original one uniformly at random with repetition. In this way, a distribution of scores can be obtained for each solver, and the solvers' statistics (mean or variance) can be compared one another using standard parametric tools. Another possible extension, would be to combine several scoring methods together, using some meta-aggregation schema, and to use the combined scores of several different procedures to compute the final score of the solver.

References

Arrow, K. J.; Sen, A. K.; and Suzumura, K., eds. 2002. *Handbook of Social Choice and Welfare*, volume 1. Elsevier.

Barrett, C. W.; de Moura, L.; and Stump, A. 2005. SMT-COMP: Satisfiability Modulo Theories Competition. In *CAV*, volume 3576 of *Lecture Notes in Computer Science*, 20–23.

Berre, D. L.; Narizzano, M.; Simon, L.; and Tacchella, A. 2004. The second QBF solvers evaluation. In *Seventh International Conference on Theory and Applications of Satisfiability Testing (SAT 2004)*, Lecture Notes in Computer Science. Springer Verlag.

Berre, D. L.; Simon, L.; and Tacchella, A. 2003. Challenges in the QBF arena: the SAT'03 evaluation of QBF solvers. In *Sixth International Conference on Theory and Applications of Satisfiability Testing (SAT 2003)*, volume 2919 of *Lecture Notes in Computer Science*. Springer Verlag.

Gelder, A. V.; Le Berre, D.; Biere, A.; Kullmann, O.; and Simon, L. 2006. Purse-Based Scoring for Comparison of Exponential-Time Programs. Unpublished draft.

Giunchiglia, E.; Narizzano, M.; and Tacchella, A. 2001. Quantified Boolean Formulas satisfiability library (QBFLIB). www.qbflib.org.

Hooker, J. N., and Vinay, V. 1995. Branching Rules for Satisfiability. *Journal of Automated Reasoning* 15:359–383.

Hooker, J. N. 1996. Testing Heuristics: We Have It All Wrong. *Journal of Heuristics* 1:33–42.

Kanji, G. 1999. *100 Statistical Tests - New Edition*. SAGE Publications.

Le Berre, D., and Simon, L. 2007. The SAT Competition. http://www.satcompetition.org [2006-6-2].

Long, D., and Fox, M. 2003. The 3rd International Planning Competition: Results and Analysis. *Artificial Intelligence Research* 20:1–59.

Lowry, R. 2006. *Concepts and Applications of Inferential Statistics*. Vassar College. Available on line at http://faculty.vassar.edu/lowry/webtext.html [2006-6-2].

Narizzano, M.; Pulina, L.; and Tacchella, A. 2006a. Competitive Evaluation of Automated Reasoning Tools: Statistical Testing and Empirical Scoring. In *First Workshop on Empirical Methods for the Analysis of Algorithms (EMAA 2006)*.

Narizzano, M.; Pulina, L.; and Tacchella, A. 2006b. The third QBF solvers comparative evaluation. *Journal on Satisfiability, Boolean Modeling and Computation* 2:145–164. Available on-line at http://jsat.ewi.tudelft.nl/.

Narizzano, M.; Pulina, L.; and Tacchella, A. 2006c. Voting Systems and Automated Reasoning: the QBFEVAL Case Study. In *1st Workshop on Computational Social Choice (COMSOC 2006)*.

Pulina, L. 2006. Empirical Evaluation of Scoring Methods. In *Proceedings of STAIRS*. Accepted contribution. Available on line at http://www.star.dist.unige.it/~pulina/paper/stairs06-Pulina.pdf [2006-5-12].

Saari, D. G. 2001. *Chaotic Elections! A Mathematician Looks at Voting*. American Mathematical Society.

Schulze, M. 2003. A New Monotonic and Clone-Independent Single-Winner Election Method. *Voting Matters* 9–19.

Sutcliffe, G., and Suttner, C. 2007. The CADE ATP System Competition. http://www.cs.miami.edu/~tptp/CASC [2006-6-2].

van Dongen, M. 2005. Introduction to the Solver Competition. In *CPAI 2005 proceedings*.