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

Cristiano Ghersi, Luca Pulina, Armando Tacchella

Machine Intelligence for the Diagnosis of Complex Systems

Systems and Technologies for Automated Reasoning

DIST - University of Genoa
Why running a competition is such a (big) deal?

Seemingly tiny problems which will indeed drive you crazy

Input/Output formats

Choosing the problem instances

Running the systems

Interacting with the developers

Reporting the results

...
Why running a competition is a such a (big) deal?

- Seemingly tiny problems which will indeed drive you crazy
Why running a competition is a such a (big) deal?

- Seemingly tiny problems which will indeed drive you crazy
  - Input/Output formats
  - Choosing the problem instances
  - Running the systems
  - Interacting with the developers
  - Reporting the results
  - ...

Not exactly your favorite experimental setup either

Proper experimental design is not that easy

It is systems you are comparing, not algorithms

The runtime distributions of the underlying algorithms are unknown, or if they are known, they are probably ill-behaved
Why running a competition is a such a (big) deal?

- Seemingly tiny problems which will indeed drive you crazy
  - Input/Output formats
  - Choosing the problem instances
  - Running the systems
  - Interacting with the developers
  - Reporting the results
  - ...

- Not exactly your favorite experimental setup either
Why running a competition is a such a (big) deal?

- Seemingly tiny problems which will indeed drive you crazy
  - Input/Output formats
  - Choosing the problem instances
  - Running the systems
  - Interacting with the developers
  - Reporting the results
  - ...

- Not exactly your favorite experimental setup either
  - Proper experimental design is not that easy
  - It is systems you are comparing, not algorithms
Why running a competition is a such a (big) deal?

- Seemingly tiny problems which will indeed drive you crazy
  - Input/Output formats
  - Choosing the problem instances
  - Running the systems
  - Interacting with the developers
  - Reporting the results
  - ...

- Not exactly your favorite experimental setup either
  - Proper experimental design is not that easy
  - It is systems you are comparing, not algorithms
  - The runtime distributions of the underlying algorithms are unknown, or if they are known, they are probably ill-behaved
What this presentation is NOT about

Everything you need to know before running a competition...
What this presentation is NOT about

Everything you need to know before running a competition...
... otherwise you will not run any for scheduling systems!
What this presentation is about

Which system should I buy?*

Even if a systems competition is (mostly) an ill-posed experiment, we would like to

- rank the systems to reflect their **true relative merit**, and
- know how much **confidence** we can have in the results

---

Our contributions (still ongoing work)

- Research about **ranking and reputation** (RaRe) systems
  - investigating different aggregation procedures
  - using statistical testing to validate the results
- An **in-depth account** of QBFEVAL’05 results using both aggregation procedures and statistical testing
Outline

1. The case study
   - QBFEVAL’05 dataset
   - Working hypotheses

2. RaRe systems
   - State-of-the-art
   - Yet another scoring method (YASM)
   - Comparing aggregation procedures

3. Statistical testing
   - Modelling QBFEVAL’05
   - Experimental results
What is a quantified Boolean Formula?

Consider a Boolean formula, e.g.,

\[(x_1 \lor x_2) \land (\neg x_1 \lor x_2)\]

Adding existential “∃” and universal “∀” quantifiers, e.g.,

\[\forall x_1 \exists x_2 (x_1 \lor x_2) \land (\neg x_1 \lor x_2)\]

yields a quantified Boolean formula (QBF).
What is the meaning of a QBF?

A QBF, e.g.,

$$\forall x_1 \exists x_2 (x_1 \lor x_2) \land (\neg x_1 \lor x_2)$$

is true if and only if

for every value of $x_1$ there exist a value of $x_2$ such that

$$(x_1 \lor x_2) \land (\neg x_1 \lor x_2)$$

is propositionally satisfiable

Given any QBF $\psi$:

- if $\psi = \forall x \varphi$ then $\psi$ is true iff $\varphi|_{x=0} \land \varphi|_{x=1}$ is true
- if $\psi = \exists x \varphi$ then $\psi$ is true iff $\varphi|_{x=0} \lor \varphi|_{x=1}$ is true
Some details about QBFEVAL’05

- 8 solvers on 551 instances
Some details about QBFEVAL’05

- 8 solvers on 551 instances
- Resource constraints
  - time limit: 900s (15 minutes)
  - memory limit: 900MB
Some details about QBFEVAL’05

- **8 solvers on 551 instances**
- **Resource constraints**
  - time limit: **900s** (15 minutes)
  - memory limit: **900MB**
- **The dataset has 4408 entries** with four attributes
  - SOLVER, the name of the solver
  - INSTANCE, the name of the instance
  - RESULT, one of {SAT, UNSAT, TIME, FAIL}
  - CPUTIME, the amount of CPU time consumed
Some details about QBFEVAL’05

- **8 solvers on 551 instances**
- **Resource constraints**
  - time limit: **900s** (15 minutes)
  - memory limit: **900MB**
- **The dataset has 4408 entries** with four attributes
  - SOLVER, the name of the solver
  - INSTANCE, the name of the instance
  - RESULT, one of {SAT, UNSAT, TIME, FAIL}
  - CPUBTIME, the amount of CPU time consumed
- **TIME** means that the time limit was **exceeded**
- **FAIL** is a **catchall** for any ill behaviour
Factors that we disregarded

- Memory consumption
  - Difficult to define precisely
  - Difficult to measure precisely
Factors that we disregarded

- Memory consumption
  - Difficult to define precisely
  - Difficult to measure precisely

- Correctness of the solution
  - Solving QBFs is a PSPACE-complete problem
  - The witness is not guaranteed to be compact
  - At the time, none of the solvers output a reliable witness
Factors that we disregarded

- Memory consumption
  - Difficult to define precisely
  - Difficult to measure precisely

- Correctness of the solution
  - Solving QBFs is a PSPACE-complete problem
  - The witness is not guaranteed to be compact
  - At the time, none of the solvers output a reliable witness

- Quality of the solution
  - No witness to check for quality
  - Checking could be expensive

- Noise in CPU time measures
What about CPU time?

CPU TIME MEASURED OVER 100 runs

- ESTIMATE
- NORMAL

CPU time

Sample Quantiles

Theoretical Quantiles
What about CPU time?

Noise does affect the CPU time measures of systems (statistical methods can deal with this phenomenon)
Aggregation procedures: systems contests

**CASC**  In the CADE ATP systems comparison
- solvers are ranked according to the number of times that RESULT is one of \{SAT, UNSAT\}, and
- ties are broken using average CPUTIME.

**QBFEVAL**  (before 2006) Same as CASC, but ties are broken using total CPUTIME.

**SATCOMP**  The 2005 SAT competition assigned two purses to each instance
- a solution purse, distributed uniformly, and
- a speed purse, distributed proportionally (w.r.t. speed)
among all the solvers that solve it.
A series purse is distributed to all the solvers that solve at least one instance in a series.
Aggregation procedures: voting systems

**Borda count**  Given $n$ solvers, instance $i$ ranks solver $s$ in position $p_{s,i}$ ($1 \leq p_{s,i} \leq n$). The score of $s$ is $S_{s,i} = n - p_{s,i}$.

**Range voting**  Similar to Borda count, whereas an arbitrary scale is used to associate a weight $w_p$ with each of the $n$ positions.

**Schulze’s method**  It is a Condorcet method that computes the Schwartz set to determine a winner. We use an extension of the single overall winner procedure, in order to make it capable of generating an overall ranking.
YASM: the formula

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

where:

- \( S_{s,i} \) is the score of solver \( s \) on instance \( i \)
- \( k_{s,i} \) is the Borda weight
- \( H_i \) is the instance hardness
- \( T_{s,i} \) is the solve time of solver \( s \) on instance \( i \)
- \( M_i \) is the minimum solve time among all solvers on instance \( i \)

\[
H_i = 1 - \frac{\# \text{ solvers that solved } i}{\# \text{ solvers that didn’t solve } i}
\]

\[
M_i = \min_s \{ T_{s,i} \}
\]
YASM: rationale

**What makes for a good solver?**

The ability to solve:

- many instances within the time limit \((L - T_{s,i})\)
- preferably hard ones \((1 + H_i)\)
- in a relatively short time \((\frac{L-T_{s,i}}{L-M_i})\)

**Why the Borda weight \(k_{s,i}\)?**

It helps to **stabilize** YASM against bias in the test set!
Measures to compare scoring methods

**Fidelity**  How much a scoring method reflects the true relative merits of the competitors

**Stability**  with respect to

- decreasing time limit (DTL-stability)
- decreasing test set cardinality (RDT-stability)
- biased test set (SBT-stability)
Homogeneity

- Degree of \textit{(dis)agreement} between different aggregation procedures.
Homogeneity

- Degree of **disagreement** between different aggregation procedures.
- Verify that the aggregation procedures considered
  - do not produce exactly the same solver rankings
  - do not yield antithetic solver rankings
Homogeneity

- Degree of **(dis)agreement** between different aggregation procedures.
- Verify that the aggregation procedures considered
  - do not produce exactly the same solver rankings
  - do not yield antithetic solver rankings
- Kendall rank correlation coefficient $\tau$ as measure of homogeneity.
### Homogeneity

<table>
<thead>
<tr>
<th></th>
<th>CASC</th>
<th>QBF</th>
<th>SAT</th>
<th>YASM</th>
<th>YASMv2</th>
<th>Borda</th>
<th>r.v.</th>
<th>Schulze</th>
</tr>
</thead>
<tbody>
<tr>
<td>CASC</td>
<td>–</td>
<td>1</td>
<td>0.71</td>
<td>0.86</td>
<td>0.79</td>
<td>0.86</td>
<td>0.71</td>
<td>0.86</td>
</tr>
<tr>
<td>QBF</td>
<td>–</td>
<td>0.71</td>
<td>0.86</td>
<td>0.79</td>
<td>0.86</td>
<td>0.71</td>
<td>0.86</td>
<td></td>
</tr>
<tr>
<td>SAT</td>
<td>–</td>
<td>0.86</td>
<td>0.86</td>
<td>0.71</td>
<td>0.71</td>
<td>0.71</td>
<td>0.71</td>
<td></td>
</tr>
<tr>
<td>YASM</td>
<td>–</td>
<td>–</td>
<td>0.86</td>
<td>0.71</td>
<td>0.71</td>
<td>0.71</td>
<td>0.71</td>
<td></td>
</tr>
<tr>
<td>YASMv2</td>
<td>–</td>
<td>–</td>
<td>–</td>
<td>0.86</td>
<td>0.86</td>
<td>0.86</td>
<td>0.86</td>
<td></td>
</tr>
<tr>
<td>Borda</td>
<td>–</td>
<td>–</td>
<td>–</td>
<td>0.86</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>r. v.</td>
<td>–</td>
<td>–</td>
<td>–</td>
<td>0.86</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Schulze</td>
<td>–</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

r.v. = range voting
Fidelity

- Given a **synthesized set** of raw data, evaluates whether an aggregation procedure **distorts** the results.
Fidelity

- Given a synthesized set of raw data, evaluates whether an aggregation procedure distorts the results.
- Several samples of table RUNS filled with random results:
  - RESULT is assigned to SAT/UNSAT, TIME or FAIL with equal probability
  - a value of CPU TIME is chosen uniformly at random in the interval [0;1]
**Fidelity**

- Given a **synthesized set** of raw data, evaluates whether an aggregation procedure **distorts** the results.
- Several samples of table `RUNS` filled with random results:
  - `RESULT` is assigned to `SAT/UNSAT`, `TIME` or `FAIL` with equal probability
  - a value of `CPUTIME` is chosen uniformly at random in the interval `[0;1]`
- A high-fidelity aggregation procedure:
  - computes approximately **the same scores** for each solver
  - produces a final ranking where scores have a **small variance-to-mean** ratio
### Fidelity

<table>
<thead>
<tr>
<th>Method</th>
<th>Mean</th>
<th>Std</th>
<th>Median</th>
<th>Min</th>
<th>Max</th>
<th>IQ Range</th>
<th>F</th>
</tr>
</thead>
<tbody>
<tr>
<td>QBF</td>
<td>182.25</td>
<td>7.53</td>
<td>183</td>
<td>170</td>
<td>192</td>
<td>13</td>
<td>88.54</td>
</tr>
<tr>
<td>CASC</td>
<td>182.25</td>
<td>7.53</td>
<td>183</td>
<td>170</td>
<td>192</td>
<td>13</td>
<td>88.54</td>
</tr>
<tr>
<td>SAT</td>
<td>87250</td>
<td>12520.2</td>
<td>83262.33</td>
<td>78532.74</td>
<td>119780.48</td>
<td>4263.94</td>
<td>65.56</td>
</tr>
<tr>
<td>YASM</td>
<td>46.64</td>
<td>2.22</td>
<td>46.33</td>
<td>43.56</td>
<td>51.02</td>
<td>2.82</td>
<td>85.38</td>
</tr>
<tr>
<td>YASMv2</td>
<td>1257.29</td>
<td>45.39</td>
<td>1268.73</td>
<td>1198.43</td>
<td>1312.72</td>
<td>95.11</td>
<td>91.29</td>
</tr>
<tr>
<td>Borda</td>
<td>984.5</td>
<td>127.39</td>
<td>982.5</td>
<td>752</td>
<td>1176</td>
<td>194.5</td>
<td>63.95</td>
</tr>
<tr>
<td>r. v.</td>
<td>12010.25</td>
<td>5183.86</td>
<td>12104</td>
<td>5186</td>
<td>21504</td>
<td>8096</td>
<td>24.12</td>
</tr>
<tr>
<td>SCHULZE</td>
<td>–</td>
<td>–</td>
<td>–</td>
<td>–</td>
<td>–</td>
<td>–</td>
<td>–</td>
</tr>
</tbody>
</table>

r.v. = range voting
RDT-stability

- Stability on a **Randomized Decreasing Test set** aims to measure how much an aggregation procedure is sensitive to perturbations that diminish the size of the original test set.
RDT-stability

- Stability on a **Randomized Decreasing Test set** aims to measure how much an aggregation procedure is sensitive to perturbations that diminish the size of the original test set.

<table>
<thead>
<tr>
<th>INSTANCE_1</th>
</tr>
</thead>
<tbody>
<tr>
<td>INSTANCE_2</td>
</tr>
<tr>
<td>INSTANCE_3</td>
</tr>
<tr>
<td>INSTANCE_4</td>
</tr>
<tr>
<td>INSTANCE_5</td>
</tr>
<tr>
<td>INSTANCE_6</td>
</tr>
<tr>
<td>INSTANCE_7</td>
</tr>
<tr>
<td>INSTANCE_8</td>
</tr>
<tr>
<td>INSTANCE_9</td>
</tr>
<tr>
<td>INSTANCE_10</td>
</tr>
<tr>
<td>INSTANCE_11</td>
</tr>
<tr>
<td>INSTANCE_12</td>
</tr>
<tr>
<td>INSTANCE_13</td>
</tr>
<tr>
<td>INSTANCE_14</td>
</tr>
<tr>
<td>INSTANCE_15</td>
</tr>
</tbody>
</table>
RDT-stability

- Stability on a **Randomized Decreasing Test set** aims to measure how much an aggregation procedure is sensitive to perturbations that diminish the size of the original test set.
RDT-stability

Stability on a **Randomized Decreasing Test set** aims to measure how much an aggregation procedure is sensitive to perturbations that diminish the size of the original test set.

```
INSTANCE_1
INSTANCE_3
INSTANCE_6
INSTANCE_7
INSTANCE_8
INSTANCE_9
INSTANCE_11
INSTANCE_12
INSTANCE_14
INSTANCE_15
```
Stability on a **Randomized Decreasing Test set** aims to measure how much an aggregation procedure is sensitive to perturbations that diminish the size of the original test set.

\[
\text{INSTANCE}_1 \quad \rightarrow \quad \text{RANKING}_A
\]

\[
\text{INSTANCE}_2 \quad \text{INSTANCE}_3 \\
\text{INSTANCE}_4 \\
\text{INSTANCE}_5 \\
\text{INSTANCE}_6 \quad \text{INSTANCE}_7 \\
\text{INSTANCE}_8 \quad \text{INSTANCE}_9 \\
\text{INSTANCE}_{10} \quad \text{INSTANCE}_{11} \\
\text{INSTANCE}_{12} \\
\text{INSTANCE}_{13} \quad \text{INSTANCE}_{14} \\
\text{INSTANCE}_{15}
\]
RDT-stability

- Stability on a **Randomized Decreasing Test set** aims to measure how much an aggregation procedure is sensitive to perturbations that diminish the size of the original test set.

```
INSTANCE_1
INSTANCE_2
INSTANCE_3
INSTANCE_4

INSTANCE_6
INSTANCE_7

INSTANCE_10
INSTANCE_11
INSTANCE_12
INSTANCE_13
```

→  RANKING_A
Stability on a **Randomized Decreasing Test set** aims to measure how much an aggregation procedure is sensitive to perturbations that diminish the size of the original test set.

- INSTANCE_1
- INSTANCE_2
- INSTANCE_3
- INSTANCE_4
- INSTANCE_6
- INSTANCE_7
- INSTANCE_10
- INSTANCE_11
- INSTANCE_12
- INSTANCE_13

→ RANKING_A
RANKING_B
Stability on a **Randomized Decreasing Test set** aims to measure how much an aggregation procedure is sensitive to perturbations that diminish the size of the original test set.
RDT-stability

- Stability on a **Randomized Decreasing Test set** aims to measure how much an aggregation procedure is sensitive to perturbations that diminish the size of the original test set.

```
INST_2
INST_3
INST_4
INST_5
INST_6

INST_8  →  RANK_A
INST_9
INST_10

INST_14
INST_15
```

Armando Tacchella  
SSC 2007 - Providence - September 22, 2007
RDT-stability

- Stability on a **Randomized Decreasing Test set** aims to measure how much an aggregation procedure is sensitive to perturbations that diminish the size of the original test set.
RD-T-stability

- Stability on a **Randomized Decreasing Test set** aims to measure how much an aggregation procedure is sensitive to perturbations that diminish the size of the original test set.

```
INSTANCE_2  INSTANCE_3  INSTANCE_4  INSTANCE_5  INSTANCE_6
INSTANCE_8  INSTANCE_9  INSTANCE_10
INSTANCE_14  INSTANCE_15

→  RANKING_A  RANKING_B  RANKING_C  →  RANKING_MEDIAN
```
RDT-stability

RDT-stability of CASC aggregation procedure

- QUANTOR
- semprop
- yQuaffle
- sSolve
- QMRes
- openQbf
- WalkQSAT
- qbfbdd

Armando Tacchella
SSC 2007 - Providence - September 22, 2007
The case study
RaRe systems
Statistical testing
State-of-the-art
Yet another scoring method (YASM)
Comparing aggregation procedures

RDT-stability

CASC
SAT
YASv2

Borda
r.v.
Schulze
DTL-stability

Stability on a **Decreasing Time Limit** aims to measure how much an aggregation procedure is sensitive to perturbations that diminish the maximum amount of CPU time granted to the solvers.
Stability on a **Decreasing Time Limit** aims to measure how much an aggregation procedure is sensitive to perturbations that diminish the maximum amount of CPU time granted to the solvers.
Stability on a **Decreasing Time Limit** aims to measure how much an aggregation procedure is sensitive to perturbations that diminish the maximum amount of CPU time granted to the solvers.
Stability on a **Decreasing Time Limit** aims to measure how much an aggregation procedure is sensitive to perturbations that diminish the maximum amount of CPU time granted to the solvers.

Armando Tacchella
Stability on a **Decreasing Time Limit** aims to measure how much an aggregation procedure is sensitive to perturbations that diminish the maximum amount of CPU time granted to the solvers.
DTL-stability

The case study
RaRe systems
Statistical testing
State-of-the-art
Yet another scoring method (YASM)
Comparing aggregation procedures

YASMv2

Armando Tacchella
SSC 2007 - Providence - September 22, 2007
The case study
RaRe systems
Statistical testing

State-of-the-art
Yet another scoring method (YASM)
Comparing aggregation procedures

DTL-stability

CASC

Borda
SBT-stability

Stability on a **Solver Biased Test set** aims to measure how much an aggregation procedure is sensitive to a test set that is biased in favor of a given solver.
SBT-stability

Stability on a **Solver Biased Test set** aims to measure how much an aggregation procedure is sensitive to a test set that is biased in favor of a given solver.

- Test set instances
- Solved by SOLVER_1
- Solved by SOLVER_2
- Solved by SOLVER_3
SBT-stability

Stability on a **Solver Biased Test set** aims to measure how much an aggregation procedure is sensitive to a test set that is biased in favor of a given solver.

- Test set instances
  - Solved by SOLVER_1
  - Solved by SOLVER_2
  - Solved by SOLVER_3
Stability on a **Solver Biased Test set** aims to measure how much an aggregation procedure is sensitive to a test set that is biased in favor of a given solver.
SBT-stability

- Stability on a **Solver Biased Test set** aims to measure how much an aggregation procedure is sensitive to a test set that is biased in favor of a given solver.
SBT-stability

YASMv2
## SBT-stability

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

Kendall $\tau$ between rankings on biased test sets (rows) vs. the original one (columns)
Null and alternative hypotheses

- We are interested in statistically significant differences in the (average) performances of the solvers
We are interested in statistically significant differences in the (average) performances of the solvers. Given any two solvers A and B we 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 performances of B.
Two fundamental issues

Let $X_A$ and $X_B$ be the vectors of run times for solvers $A$ and $B$

1. How do we consider TIME and FAIL values in $X_A$ and $X_B$?

2. Which assumptions, if any, can be made about the underlying distributions of $X_A$ and $X_B$?
FAT (Failure as time limit) FAIL is replaced by TIME
- Consistently overestimates the performances of the solvers, but
- it allows the paired comparison of the values in $X_A$ and in $X_B$. 
Data models

FAT (Failure as time limit) FAIL is replaced by TIME
- Consistently overestimates the performances of the solvers, but
- it allows the paired comparison of the values in $X_A$ and in $X_B$.

TAF (Time limit as failure) TIME is replaced by FAIL and both are considered "missing values"
- Overestimation does not occur, but
- $X_A$ and $X_B$ may not be equal in length, so their paired comparison is not generally possible.
Parametric or distribution-free?

For each solver $A$

- we check $X_A$ under FAT and TAF models using
- the Shapiro-Wilk test of the null hypothesis that the samples come from a *normally distributed* population.
For each solver $A$ we check $X_A$ under FAT and TAF models using the **Shapiro-Wilk** test of the null hypothesis that the samples come from a **normally distributed** population.

<table>
<thead>
<tr>
<th>$X_A$</th>
<th>FAT</th>
<th>TAF</th>
</tr>
</thead>
<tbody>
<tr>
<td>OPENQBF</td>
<td>$9.665 \times 10^{27}$</td>
<td>$2.036 \times 10^{24}$</td>
</tr>
<tr>
<td>QBFBDD</td>
<td>$2.768 \times 10^{30}$</td>
<td>$7.051 \times 10^{19}$</td>
</tr>
<tr>
<td>QMRES</td>
<td>$1.419 \times 10^{27}$</td>
<td>$1.588 \times 10^{28}$</td>
</tr>
<tr>
<td>QUANTOR</td>
<td>$8.334 \times 10^{32}$</td>
<td>$6.926 \times 10^{36}$</td>
</tr>
<tr>
<td>SEMPROP</td>
<td>$5.012 \times 10^{29}$</td>
<td>$2.359 \times 10^{31}$</td>
</tr>
<tr>
<td>SSOLVE</td>
<td>$9.513 \times 10^{28}$</td>
<td>$1.359 \times 10^{29}$</td>
</tr>
<tr>
<td>WALKQSAT</td>
<td>$1.148 \times 10^{27}$</td>
<td>$6.414 \times 10^{27}$</td>
</tr>
<tr>
<td>YQUAFFLE</td>
<td>$6.753 \times 10^{28}$</td>
<td>$5.453 \times 10^{30}$</td>
</tr>
</tbody>
</table>

(Values: Shapiro-Wilk test *p*-values)
For each solver $A$, we check $X_A$ under FAT and TAF models using the Shapiro-Wilk test of the null hypothesis that the samples come from a normally distributed population.

<table>
<thead>
<tr>
<th>$X_A$</th>
<th>FAT</th>
<th>TAF</th>
</tr>
</thead>
<tbody>
<tr>
<td>OPENQBF</td>
<td>$9.665 \times 10^{27}$</td>
<td>$2.036 \times 10^{24}$</td>
</tr>
<tr>
<td>QBFBDD</td>
<td>$2.768 \times 10^{30}$</td>
<td>$7.051 \times 10^{19}$</td>
</tr>
<tr>
<td>QMRES</td>
<td>$1.419 \times 10^{27}$</td>
<td>$1.588 \times 10^{28}$</td>
</tr>
<tr>
<td>QUANTOR</td>
<td>$8.334 \times 10^{32}$</td>
<td>$6.926 \times 10^{36}$</td>
</tr>
<tr>
<td>SEMPROP</td>
<td>$5.012 \times 10^{29}$</td>
<td>$2.359 \times 10^{31}$</td>
</tr>
<tr>
<td>SSOLVE</td>
<td>$9.513 \times 10^{28}$</td>
<td>$1.359 \times 10^{29}$</td>
</tr>
<tr>
<td>WALKQSAT</td>
<td>$1.148 \times 10^{27}$</td>
<td>$6.414 \times 10^{27}$</td>
</tr>
<tr>
<td>YQUAFFLE</td>
<td>$6.753 \times 10^{28}$</td>
<td>$5.453 \times 10^{30}$</td>
</tr>
</tbody>
</table>

(Values: Shapiro-Wilk test $p$-values)

It is highly unlikely that the $X_A$’s are normally distributed!
Wilcoxon signed rank (WSR) test

- A distribution-free alternative to correlated-samples t-test
- \( H_0 \) is that \( X_A \) and \( X_B \) do not differ significantly (on average)
- Its basic assumptions are
  - that the paired values of \( X_A \) and \( X_B \) are randomly and independently drawn;
  - that the dependent variable is intrinsically continuous; and
  - that the measures of \( X_A \) and \( X_B \) have the properties of at least an ordinal scale of measurement.

WSR test is ok with the FAT model, but not with the TAF one!
QBFEVAL’05 dataset and the WSR test

- Nodes correspond to solvers
- An edge from $A$ to $B$ means
  \[
  \frac{\text{# of times } (X_A - X_B) > 0}{\text{# of times } (X_B - X_A) > 0} > 1
  \]
- A path between $A$ and $B$ means that WSR rejects $H_0$
  - Confidence level: 99%
  - Control: family-wise error rate
Mann-Whitney-Wilcoxon (MWW) test

- A distribution-free alternative to independent-samples t-test
- $H_0$ is that $X_A$ and $X_B$ do not differ substantially
- Its basic assumptions are
  - that $X_A$ and $X_B$ are randomly and independently drawn;
  - that the dependent variable is intrinsically continuous; and
  - that the measures of $X_A$ and $X_B$ have the properties of at least an ordinal scale of measurement.

MWW test is ok with the TAF model, and it gives an approximate, although conservative, picture.
QBFEVAL’05 dataset and the MWW test

- Nodes correspond to solvers
- An edge from A to B means
  \[
  \frac{\# \text{ of times } (X_A - X_B) > 0}{\# \text{ of times } (X_B - X_A) > 0} > 1
  \]
  under the FAT model.
- A path between A and B means that MWW rejects \( H_0 \)
  - Confidence level: 99%
  - Control: family-wise error rate
under the TAF model.
All the scoring methods produce rankings mostly compatible with WSR and MWW although
- SAT conflicts with WSR on QMRES vs. SEMPROP, but
- MWW finds the two incomparable.
All the scoring methods produce rankings mostly compatible with WSR and MWW although
- SAT conflicts with WSR on QMRES vs. SEMPROP, but
- MWW finds the two incomparable.

QMRES, SSOLVE and YQUAFFLE are
- incomparable according to WSR, and
- the solvers on which the rankings mostly differ.
Scoring methods, WSR and MWW (1/2)

- All the scoring methods produce rankings mostly compatible with WSR and MWW although
  - SAT conflicts with WSR on QMRES vs. SEMPROP, but
  - MWW finds the two incomparable.
- QMRES, SSOLVE and YQUAFFLE are
  - incomparable according to WSR, and
  - the solvers on which the rankings mostly differ.
- MWW finds also
  - SEMPROP to be incomparable w.r.t. QMRES, SSOLVE and YQUAFFLE, but
  - all the methods, except SAT, rank SEMPROP second best.
WSR and MWW rankings obtained by

- considering the DAGs induced by the two tests, and
- breaking ties in reverse order of edge labels.

<table>
<thead>
<tr>
<th></th>
<th>Borda</th>
<th>MWW</th>
<th>QBF/CASC</th>
<th>r.v.</th>
<th>SAT</th>
<th>Schulze</th>
<th>WSR</th>
</tr>
</thead>
<tbody>
<tr>
<td>MWW</td>
<td>0.93</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>QBF/CASC</td>
<td>0.84</td>
<td>0.76</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>r.v.</td>
<td>0.86</td>
<td>0.79</td>
<td>0.69</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>SAT</td>
<td>0.71</td>
<td>0.64</td>
<td>0.69</td>
<td>0.71</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Schulze</td>
<td>1.00</td>
<td>0.93</td>
<td>0.84</td>
<td>0.86</td>
<td>0.71</td>
<td>1.00</td>
<td>-</td>
</tr>
<tr>
<td>WSR</td>
<td>1.00</td>
<td>0.93</td>
<td>0.84</td>
<td>0.86</td>
<td>0.71</td>
<td>1.00</td>
<td>-</td>
</tr>
<tr>
<td>YASM</td>
<td>0.86</td>
<td>0.79</td>
<td>0.69</td>
<td>0.86</td>
<td>0.86</td>
<td>0.86</td>
<td>0.86</td>
</tr>
</tbody>
</table>

(Values: Kendall’s τ between rankings)
Empirical scoring can borrow a lot from voting theory and benefit from statistical testing.

Elaborate scoring methods are not necessarily better than simple ones.

Statistical testing provides insightful cross-validation of the empirical scoring results.

Possible extensions:

- Is there a better YASM than YASM?
- Are there other useful statistical techniques?
Measures to compare scoring methods

**Fidelity**  How much a scoring method reflects the true relative merits of the competitors
Measures to compare scoring methods

**Fidelity**  How much a scoring method reflects the **true relative merits** of the competitors

**Stability**  with respect to

- decreasing **time limit** (DTL-stability)
- decreasing test set **cardinality** (RDT-stability)
- **biased** test set (SBT-stability)
Measures to compare scoring methods

Fidelity  How much a scoring method reflects the true relative merits of the competitors

Stability  with respect to
  - decreasing time limit (DTL-stability)
  - decreasing test set cardinality (RDT-stability)
  - biased test set (SBT-stability)

SOTA distance  Considering $M_i = \min_s \{ T_{s,i} \}$ and given $m$ instances, the distance of solver $s$ from the state-of-the-art (SOTA) solver is

$$d_s = \sqrt{\sum_{i=1}^{m} (T_{s,i} - M_i)^2}$$
Feed each scoring method with “white noise”

- RESULT equally likely to be either SAT, UNSAT, TIME, or FAIL
- CPU TIME distributed uniformly in [0,1]
- generate several sample datasets accordingly

<table>
<thead>
<tr>
<th>Method</th>
<th>Median</th>
<th>Min</th>
<th>Max</th>
<th>IQ Range</th>
<th>F</th>
</tr>
</thead>
<tbody>
<tr>
<td>QBF</td>
<td>183.00</td>
<td>170.00</td>
<td>192.00</td>
<td>13.00</td>
<td>88.54</td>
</tr>
<tr>
<td>CASC</td>
<td>183.00</td>
<td>170.00</td>
<td>192.00</td>
<td>13.00</td>
<td>88.54</td>
</tr>
<tr>
<td>SAT</td>
<td>83262.33</td>
<td>78532.74</td>
<td>119780.48</td>
<td>4263.94</td>
<td>65.56</td>
</tr>
<tr>
<td>YASM</td>
<td>1268.73</td>
<td>1198.43</td>
<td>1312.72</td>
<td>95.11</td>
<td>91.29</td>
</tr>
<tr>
<td>Borda</td>
<td>982.50</td>
<td>752.00</td>
<td>1176.00</td>
<td>194.50</td>
<td>63.95</td>
</tr>
<tr>
<td>r.v.</td>
<td>12104.00</td>
<td>5186.00</td>
<td>21504.00</td>
<td>8096.00</td>
<td>24.12</td>
</tr>
</tbody>
</table>

(Values: scoring statistics over 100 random datasets)

The fidelity index F is Min/Max × 100
Given a scoring method

- obtain the ranking $R$ using the entire dataset,
- consider the ranking $R_s$ obtained by removing from the dataset all the instances that are not solved by $s$, and
- compare $R$ and $R_s$ using Kendall’s $\tau$.

<table>
<thead>
<tr>
<th></th>
<th>CASC/QBF</th>
<th>SAT</th>
<th>YASM</th>
<th>Borda</th>
<th>r.v.</th>
<th>Schulze</th>
</tr>
</thead>
<tbody>
<tr>
<td>OPENQBF</td>
<td>0.43</td>
<td>0.57</td>
<td>0.64</td>
<td>0.79</td>
<td>0.79</td>
<td>0.79</td>
</tr>
<tr>
<td>QBFBDD</td>
<td>0.43</td>
<td>0.43</td>
<td>0.64</td>
<td>0.79</td>
<td>0.86</td>
<td>0.79</td>
</tr>
<tr>
<td>QMRes</td>
<td>0.64</td>
<td>0.86</td>
<td>0.79</td>
<td>0.71</td>
<td>0.86</td>
<td>0.71</td>
</tr>
<tr>
<td>QUANTOR</td>
<td>1</td>
<td>0.86</td>
<td>0.86</td>
<td>0.93</td>
<td>0.86</td>
<td>1</td>
</tr>
<tr>
<td>SEMPROP</td>
<td>0.93</td>
<td>0.71</td>
<td>0.79</td>
<td>0.93</td>
<td>0.86</td>
<td>0.93</td>
</tr>
<tr>
<td>SSOLVE</td>
<td>0.71</td>
<td>0.57</td>
<td>0.79</td>
<td>0.86</td>
<td>0.79</td>
<td>0.86</td>
</tr>
<tr>
<td>WALKQSAT</td>
<td>0.57</td>
<td>0.57</td>
<td>0.71</td>
<td>0.64</td>
<td>0.79</td>
<td>0.71</td>
</tr>
<tr>
<td>YQUAFFLE</td>
<td>0.71</td>
<td>0.64</td>
<td>0.71</td>
<td>0.86</td>
<td>0.86</td>
<td>0.86</td>
</tr>
<tr>
<td>Mean</td>
<td>0.68</td>
<td>0.65</td>
<td>0.74</td>
<td>0.81</td>
<td>0.83</td>
<td>0.83</td>
</tr>
</tbody>
</table>
Given a scoring method

- obtain the ranking $R$ using the entire dataset,
- consider the ranking $S$ induced by the SOTA-distance, and
- compare $R$ and $S$ using Kendall’s $\tau$.

<table>
<thead>
<tr>
<th>SOTA-distance</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>CASC</td>
<td>1.00</td>
</tr>
<tr>
<td>QBF</td>
<td>1.00</td>
</tr>
<tr>
<td>SAT</td>
<td>0.71</td>
</tr>
<tr>
<td>YASM</td>
<td>0.79</td>
</tr>
<tr>
<td>Borda</td>
<td>0.86</td>
</tr>
<tr>
<td>r.v.</td>
<td>0.71</td>
</tr>
<tr>
<td>Schulze</td>
<td>0.86</td>
</tr>
</tbody>
</table>