Careful Ranking of Multiple Solvers with Timeouts and Ties

Allen Van Gelder
Computer Science Dept.
Univ. of California
Santa Cruz, CA, USA

http://www.cse.ucsc.edu/∼avg/

http://www.cse.ucsc.edu/∼avg/Papers/
These slides are careful-trans.pdf

http://www.cse.ucsc.edu/∼avg/CarefulRanking/
Software directory, contains CarefulRanking.tar
as well as individual files.

The Problem:

Compare performance of two or more programs on a set of benchmarks under conditions that not all programs will complete all benchmarks successfully, *due to resource limits.*

Applications:

- Evaluate whether a software change made a significant difference between two versions.
- Rank a set of k solvers in a competition.
  - Relative ranking of A and B should not depend on C.
  - Rankings should be stable as *arbitrary* resource limits vary.
  - Rankings should be comprehesible and easily checked for accuracy.
  - New solver can be ranked later.
  - Small timing differences can be ignored.
  - A minimum of *arbitrary* parameters.

"*Arbitrary*" means not directly related to the underlying intended uses of the programs.

The “Competition”:

Solution-count ranking (recent SAT and related competitions)

- `http://www.satcompetition.org/2009/spec2009.html`

Voting methods from social choice theory

Heavy-duty statistics, such as Nikolić in SAT 2010.

Related Work:

L. Pulina, Empirical evaluation of scoring methods,
Starting AI Researcher Symposium, 2006.

M. Nikolić, Statistical methodology for comparison of SAT solvers,
SAT 2010.

Comparison of Two Solvers

Input:

SOLVER	solver1	solver2
benchA.cnf	3.010	5.280
benchB.cnf	3233.020	578.420
benchC.cnf	893.010	Inf
benchD.cnf	Inf	Inf
	...	
benchZZ.cnf	Inf	34.242

Noise parameter: 60 seconds

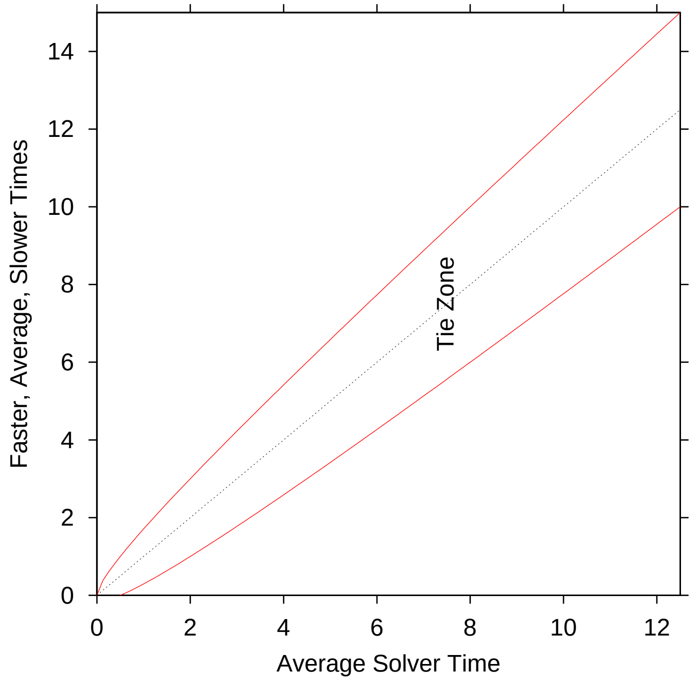
Solution-count ranking says ...

solver1 wins on benchC;
solver2 wins on benchZZ;
All others shown are ties.

Careful ranking says ...

Same wins as Solution-count ranking above;
solver2 wins on benchB (time diff. $>=$ 477).
Tie on benchA (both times "in the noise").

Careful Ranking, Tie Zone Growth with Average Solving Time



Tie Zone for **noise** = 1 minute;
times in minutes.

Lower curve:
faster solver time;

upper curve:
slower solver time;

middle line:
average of
upper and lower curves.

For benchmark i:

$$\alpha = \sqrt{\textbf{noise}/2}$$

$$avTime = \frac{(t_i(solver1) + t_i(solver2))}{2}$$

$$\Delta = \alpha\sqrt{avTime}.$$

Tie zone extends from $avTime - \Delta$ to $avTime + \Delta$.

Rationale: grows like boundaries for Poisson processes, random walk, etc.

Comparison of Two Solvers, Continued

Each benchmark is a mini-match worth ± 1 point.

The raw score is the sum over all n benchmarks.

Let n_{ties} be the number of tied mini-matches.

Null Hypothesis: Solvers are equally likely to win a random mini-match.

One *standard deviation* (σ) is about $\frac{1}{2}\sqrt{n - n_{ties}}$.

Normalized match score (approximate Student t statistic) $=$ (raw score) / σ.

Roughly speaking, 2.5 σ is significant at the 1% level, 2 σ is significant at the 5% level and 1.5 σ is significant at the 10% level.

For example, if solver1 is *not* really better than solver2, the probability that it wins a match by 2σ due to random chance is less than 0.05.

Note: Significance is *not* part of our proposal for competition ranking.

Ranking a k-Way Competition

Dominance Matrix $M[i, j] =$

1 if solver i has any positive raw score vs. solver j;
0 if $i = j$ or solver i has any negative raw score vs. solver j;
$\frac{1}{2}$ if $i \neq j$ and solver i tied solver j on raw score.

Treat M as the adjacency matrix of a directed graph.

- $\frac{1}{2}$ is considered an edge.

Preliminary rank(solver i) = $k - \sum_{j=1}^{k} M[i, j]$.

Simple case: Contiguous integer ranks $1, 2, \ldots, r$ exist, and the next rank is $> r + 1$.

- Ranks $1, 2, \ldots, r$ are the unambiguous r best solvers, in order.
- These solvers are not involved in any raw-score ties.
- The next ranked solver(s) are involved in a cycle.

Dealing with Cycles: All solvers in a strongly connected component have *equivalent rank*.

- Use M to find cycles; preliminary ranks are not reliable.
- Tie-breaks within a strongly connected component may be by total raw scores among solvers in that SCC (round robin scheme).

Example, Application Track, SAT 2009 Competition

time limit: 10000 seconds   16 solvers   final phase   prelim. inv. rank = 16− prelim. rank

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	num.	abbrev. name	prelim. inv. rank
0	0	0	0	0	0	0	1	1	1	1	1	0	0	1	0	1	CircUs	6
1	0	1	1	1	0	1	1	1	1	1	1	0	1	1	0	2	LySAT_i	12
1	0	0	0	0	0	1	1	1	1	1	1	0	1	1	0	3	MXC	9
1	0	1	0	0	0	1	1	1	1	1	1	0	1	1	0	4	ManySAT_1.1	10
1	0	1	1	0	0	1	1	1	1	1	1	0	1	1	0	5	MiniSAT_09z	11
1	1	1	1	1	0	1	1	1	1	1	1	0	1	1	0	6	MiniSat_2.1	13
1	0	0	0	0	0	0	1	1	1	1	1	0	1	1	0	7	Rsat	8
0	0	0	0	0	0	0	0	1	0	1	1	0	0	0	0	8	SAT07_Rsat	3
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	9	SAT07_picosat	0
0	0	0	0	0	0	0	1	1	0	1	1	0	0	1	0	10	SATzilla	5
0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	11	SApperloT	1
0	0	0	0	0	0	0	0	1	0	1	0	0	0	0	0	12	clasp	2
1	1	1	1	1	1	1	1	1	1	1	1	0	1	1	0	13	glucose	14
1	0	0	0	0	0	0	1	1	1	1	1	0	0	0	0	14	kw	6
0	0	0	0	0	0	0	1	1	0	1	1	0	1	0	0	15	minisat_cumr	5
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	16	precosat	15

8 contiguous preliminary inv. ranks 15, 14, 13, 12, 11, 10, 9, 8 exist.

4 contiguous preliminary inv. ranks 0, 1, 2, 3 exist.

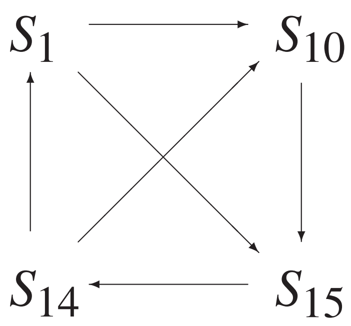
Solvers 1, 10, 14, and 15 are in one SCC, so they share inverse ranks 4–7.

Tie-Break Illustration

Recall that solvers 1, 10, 14, and 15 are in one strongly connected component, sharing ranks 9–12.
For purposes of illustration, we apply the round-robin tie-break procedure to these four solvers, although in practice it is probably not important to break this tie.

The left side just below shows the raw scores for solvers involved. The right side shows the dominance subgraph.

	S_1	S_{10}	S_{14}	S_{15}
S_1	0	13	-9	1
S_{10}	-13	0	-8	3
S_{14}	9	8	0	-1
S_{15}	-1	-3	1	0



The round-robin ranking, based on the row-sums, gives $S_{14} > S_1 > S_{15} > S_{10}$.
Proceedings typo: S_1, S_{10} switched.

Changes in Top Three Ranks as Time Limit Varied

Time limit was varied downward continuously, from 10000 seconds.

time range	solution-count	careful rank
1600–2000	8	4
2000–4000	10	0
4000–6000	4	0
6000–8000	0	0
8000–10000	1	0

Notes

(1) `precosat` stayed in first place for all time limits 4000 seconds and above, in both ranking methods.

(2) Returning to an earlier permutation is considered a rank change.

Careful ranking is more robust by this criterion.

Intuitive explanation: The Solution-Counting Paradox

> The *larger* the time-out the *lower* the discrimation of solution-counting!
> (In the limit, all solvers *tie*.)

With **solution-count ranking**, a mini-match victory is only *temporary*:

- S_1 wins the mini-match against S_2 only if S_1 succeeds and S_2 times out.
- But for a high enough time limit S_2 also succeeds (in theory), and the victory is wiped out.

With **careful ranking**, once the time limit is above the *tie zone*, and S_2 still has not succeeded, the victory is *permanent* for this mini-match.

Differences in Ranking, Part 1

The two ranking methods, **careful ranking** and **solution-count ranking**, disagreed on the third place solver with the final time limit of 10,000 seconds.

`MiniSat_2.1` held third place behind `glucose` for all time limits above 2000 under **careful ranking**.

Under the **solution-count ranking**:

- Several times, `MiniSat_2.1` and `LySAT_i` exchanged positions two and three.
- `LySAT_i` finally took the lead after about 8100 secs.
- By the 10,000 time-out, `MiniSat_2.1` was in sixth place.

Differences in Ranking, Part 2

Under the **solution-count ranking**, `precosat` and `glucose` were apparently “neck and neck,” as they each solved 204 instances.

- The tie-break was on cumulative CPU time, and `precosat` won.
- Other solvers were in the 190’s well separated from the two leaders.

Quite a different picture emerges under **careful ranking**.
We show three matches with their statistics.

Winner	Loser	Raw Score	Std. Devs.	Prob. Faster
`precosat`	`glucose`	16	1.65	0.97
`glucose`	`MiniSat_2.1`	8	0.83	0.79
`MiniSat_2.1`	`LySAT_i`	8	0.86	0.80

In this ranking, `precosat` has a more convincing win than any of the others.

Conclusion

This paper described a new ranking system that

- provides a measure of statistical significance,
- allows for small timing differences to be treated as ties,
- ensures that a pairwise comparison between two solvers is not influenced by a third solver.

The latter property also allows later researchers to replicate the competition conditions and find out where their solver would have ranked.

Another application of this technique is to evaluate whether software changes from one version to another caused a performance difference that is statistically significant, or whether the difference is in a range that might well just be random.

Acknowledgment:
Thanks to Daniel Le Berre, Laurent Simon, and Olivier Roussel
for 2009 SAT Competition data in text form.